UNIVERSITY OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Ashikur Rahman

Development and evaluation of a smartphone-based system for inspection of road maintenance work

Master’s Thesis
Degree Programme in Computer Science and Engineering
2021

Rahman A. (2021) Development and evaluation of a smartphone-based system for inspection of road maintenance work. University of Oulu, Degree Programme in Computer Science and Engineering. Master's Thesis, 74p.

ABSTRACT

In the road construction industry, doing work inspection is a laborious and resource-consuming job because of the distributed work site. Contractors in Finland require to capture photos of every road fix they have done as proof of their work. It is well-established that with the help of smartphone technology, these kinds of manual work can be reduced. This thesis aims to develop and evaluate a smartphone-based system to capture video evidence of task completion.

The system, designed and developed in this thesis, consists of an Android application named 'Road Recorder' and a web tool for managing the content collected by Road Recorder. While mounted to a vehicle's dashboard used in construction work, the Road Recorder can record the videos of road surface and geo-location information and some other metadata and send them to a remote server that is inspected using the web tool.

Users of different backgrounds were given the system to accomplish some tasks and were observed closely. The users were interviewed at the end, and responses were analyzed to find the usability of the applications. The results indicate the high usability of the Road Recorder application and reveal possible improvements for the Road Recorder management web application.

Overall, Road Recorder is a great step towards the automation of such construction work inspection. Though there were some limitations in the evaluation process, it demonstrates that Road Recorder is easy to use and can be a useful tool in the industry.

Keywords: Android, Smartphone, Road maintenance, In-vehicle system, Usability.

TABLE OF CONTENTS

FOREWORD

At first, I would like to thank my supervisor Dr. Ekaterina Gilman for her continuous support, suggestion, and dedication. She reviewed my thesis always with great patient.

I want to thank my second supervisor Associate Professor Dr. Georgi V. Georgiev, and technical supervisor Iván Sánchez Milara for their suggestions to design application tests and reviewing my thesis.

I am grateful to Timo Salminen and my second technical supervisor Antti Jokela of Ficonic Solutions Oy for allowing me to work on the topic and for continuous support and advice.

Special thanks to my family members for giving me motivation and inspiration during the thesis writing period.

Finally, my special gratitude goes to my late father, who always supported my dreams and motivated me to study abroad.

Oulu, June 9th, 2021

Ashikur Rahman

LIST OF ABBREVIATIONS AND SYMBOLS

API	Application Programming interface
CPU	Central Processing Unit
CSS	Cascading Style Sheets
ERP	Enterprise Resource Planning
GPS	Global Positioning System
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HD	High-Definition
HTML	Hypertext Mark-up Language
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
IVMS	In-Vehicle Monitoring System
REST	Representational State Transfer
SDK	Software Development Kit
UAV	Unmanned Aerial Vehicle
UBICOMP	Centre for Ubiquitous Computing
UI	User Interface
VPADS	Video-based Pavement Distress Screening

1. INTRODUCTION

Routa Systems[1] is an ERP system that is designed and developed by Ficonic Solutions Oy[2] especially for infrastructure maintenance companies. The focus of Routa Systems is road maintenance work management, but the system is easily customizable to use in any other infrastructure development and maintenance management. The system helps organize and digitize different aspects of road construction and maintenance, including but not limited to real-time tracking of the vehicle and road work, navigation, reporting road conditions, maintaining the whole process of the work, and employee's work time management.

Road Recorder, developed in this thesis, is an in-vehicle monitoring system that is a part of the Routa Systems. Road recorder is designed and developed to be used in Android smartphones mounted in vehicles. The Road Recorder application automatically records videos of the roads when vehicles are in work mode and stores them to help inspect the road condition and keep the workers' proof of road works. Road Recorder also has a web-based application for managing content created by the Road Recorder Android application.

1.1. Motivation

In recent years, digitalization is everywhere. Almost all the aspects of life are now to some extent involved at least some digital devices and technologies. At work, people use different kinds of digital methods, software, and tools to make the work faster and efficient. As people do more work and faster than before, the need to document their work is increasing. Almost all work needs documentation either in the form of written documents, photographs, or video, etc.

Like private organizations, government and local authorities now need more documentation of work than ever before. These documentations can be used as a reference, proof of work, or any other further investigation. With currently available software tools, it is now very easy to document works done on a desk. However, when it is about works done outside on the street, there are not many tools available to help with that.

When doing the road maintenance work, a client of Ficonic solutions was required by the authority to take photos of the road damages before and after the maintenance. A dedicated person was appointed at every worksite to take photos, which needed more money, resources, and time to manage the documentation. Moreover, storing the taken picture with necessary metadata, such as time, location, and work orders were very hectic and time-consuming. Someone from the local authority then checks these photos to ensure that maintenance work was done properly. Going through all these photos one by one and comparing the before and after photos were also a tough and time-consuming job. Moreover, it was impossible to search for specific images with metadata if needed. Besides, there was always the risk of losing these valuable photos.

Routa is one of the fastest-growing infrastructure maintenance tools in Finland. Routa Systems make the management, execution, and reporting of infrastructure

---

[1] https://routasystems.com
[2] https://www.sitowise.com (Ficonic merged into Sitowise)

construction and maintenance work more efficient. As the client couldn't find a system that could solve the problem problems they were having with their documentation, Ficonic developed the 'Road Recorder' application as a part of the Routa Systems, which solved the aforementioned problems, also saves a great deal of time, money and resources for our client. The system is very promising to be used in many other similar clients. Companies from all over Europe are showing their interest in it.

Many reasons influenced the thesis author's motivations to do a thesis on Road Recorder. When the first requirements for the app came, it was interesting for the author because the author had been actively involved with the development of Routa Systems. As an Android development enthusiast, also because the app was unique to its purpose, the author got motivated to develop the application. Moreover, the author was fascinated by the success of Routa Systems. Finally, Road Recorder can lead to future scope of contribution in digital documentation and better infrastructure maintenance systems.

1.2. Objective and scope

The main objective of this thesis is to develop the Road Recorder tool, which consists of an Android application to record video and metadata of road surfaces and a web tool for the management of the video and video metadata. This thesis will also provide a comprehensive background study and related work analysis. After the development work, there was a comprehensive test to find out the problems with the system, validate requirements, and find out what users expect more from the system, which will be presented at the end. The following tasks are done to complete the development of the Road Recorder application.

1. Requirement analysis
2. Designing the application
3. Server setup
4. Backend implementation
5. Road Recorder Android app development
6. Content management web tool development for Road Recorder
7. Testing and validating the system
8. Evaluation of the system

The target audience of the thesis is the researchers and software developers who are working with in-vehicle monitoring systems and digital documentation, also the clients of Routa Systems and other infrastructure maintenance companies.

The thesis author’s contribution to the development work was in Android application development, backend development, and testing. For the web content management tool and server setup, the author relied on his colleagues.

1.3. Structure

Chapter 1 presents the topic for this thesis. The motivations of this thesis are described. Also, the objective and scope of the research are defined here.

Chapter 2 presents the background of the thesis. Related academic and industry works are analyzed to justify the importance of the thesis.

Chapter 3 describes different tool components of Routa Systems that are related to the thesis work.

Chapter 4 and 5 presents the development of inspection tools. Chapter 4 describes the planning and designing of the application. It also describes the software architecture of the system. Chapter 5 presents how the actual implementation of the system is done, including how different applications in the system communicate with each other and work synchronously.

While chapter 6 describes the evaluation strategies of the application and the qualitative and quantitative findings from the evaluation, chapter 7 briefly presents the analysis of the testing results.

Chapter 8 suggests some improvements and future works based on the analysis of chapter 7. Chapter 9 briefly summarizes the whole thesis, and the last chapter lists all the references used in the thesis.

2. BACKGROUND

This chapter primes the thesis topics and presents the theoretical part of the thesis. It provides an overview of the related work and introduces the related concepts to help understand the bigger picture of the thesis. The chapter starts with the discussion on Road maintenance work inspection tools. Video surveillance camera systems are then described with their uses. Then In-vehicle monitoring systems are presented, along with the possibility of using a smartphone in an In-vehicle monitoring system. Finally, testing approaches are discussed.

2.1. Road maintenance inspection tools

A routine inspection in a construction site is important for many reasons. First, it ensures that the project is progressing as expected. Secondly, it guarantees that project requirements and standards are maintained properly. The inspection also helps to estimate the necessary materials, equipment, and resources needed for the work. Like any other construction site, Road maintenance work also needs a regular inspection to ensure quality.

2.1.1. The traditional way of road maintenance inspection

Traditional monitoring and inspection work in construction sites rely on lots of manual work [1]. There is always a need for a human to handle the monitoring task. The person must go through all the necessary parts of the construction and take notes on that. This involves human judgment, high costs, and is too infrequent to provide authorities timely and accurate control data [2]. Also, the quality and integrity of the data are low because of irregular data generation [1]. Traditional road construction inspection is just like any other construction work. On a road construction project, there will be inspectors who inspect if work is being done according to specifications. Inspectors travel to all the points where construction works are done and then verify that it is done properly according to company standards. Also, they note some necessary information that is used as proof of work later. Moreover, road inspection requires more resources and time to inspect than any other construction work because road work is distributed to a large area in most cases.

2.1.2. Modern road maintenance inspection tools

To overcome the limitations of manual inspection, researchers have been working on automated tools. To the best of our knowledge, no papers exist in the computer vision literature on road maintenance inspection tools. However, work exists on various related topics.

Different kinds of tools and techniques are proposed for work progress tracking. The first approach is work progress tracking by using static image processing taken by some fixed cameras. Ibrahim et al. presented a conceptual framework to automate both the planning and data collection aspects of progress monitoring of a building construction work [2]. They proposed an integrated system for progress monitoring based on the work breakdown structure (WBS), where the project work is divided into smaller manageable components [2]. This was done in two ways. Firstly, they integrated a means of modeling and assigned each work component to work packages based on precise criteria. Secondly, on-site cameras are used to take images of the site. These images are then analyzed automatically to measure the completion of components and to assess the material needed. This is combinedly used to determine work progress. Though the camera system works in a controlled situation, it has some limitations. One of the biggest concerns is the coverage of the fixed-camera approach. Components that are not in the camera scene do not necessarily mean the component is completed; hence it might need a physical inspection to verify the situation. However, the system can be improved by using multiple sources of the image i.e. multiple cameras with additional information.

One way to solve the unavailability of fixed cameras in place is to use a moving camera capable of remote sensing. With the rapid improvement of unmanned aerial vehicles (UAV), remote sensing is becoming easier. The main advantage of using UAVs is their easy access in hard-to-reach locations. UAVs can have different sensors, including, but not limited to, cameras, visible light, light detectors, ranging sensors, etc. [3]. Freimuth et al. [3] proposed a concept of doing visual inspection tasks of construction with UAVs. They developed an application that allows planning inspections in a 3D environment. The application consequently creates collision-free flight ways based on Building Information Modelling (BIM) data. When traveling through those paths, UAVs collect photographs of the resulting inspection and store them for further processing. While the system can gather rich information of the point of interest, it showed some technical problems during the pilot test, i.e. the UAV needed to maintain a fixed distance to the structure manually, which authors are planning to improve in the next phase. Quater et al. [4] used UAVs to inspect solar panels in a photovoltaic plant. Solar panel modules need regular inspections to check for possible degradation and damage. The inspection was done in two phases. UAVs were used to scan the module from high altitude in the first phase to identify damage and hot spots. In the second phase, the modules are scanned from low altitude to identify mechanical and deterioration-induced problems. While with the manual inspection, each module needs to be inspected separately, UAVs can inspect the whole plant with one single flight; hence such operation is cost-effective.

Complex megastructures like bridges also need regular inspection. Most of these inspections are mainly visual. A structure, like a bridge, may have many parts hard to reach underneath and on top of the structure. UAVs can easily be used for inspection in such cases. Hallermann et al. [5] proposed a method of visual inspection using remote control UAVs. They developed the prototype of a semi-autonomous inspection method that can detect damage automatically by analyzing photos and videos. The UAVs used in the system are GPS-controlled but can be handled manually during bad weather conditions. Geo-locations are stored with photos and videos as metadata, which can generate a 3-D model of the structure.

As can be seen, the technology aims to help to conduct routine inspections in a construction site. Since this thesis is focused on road maintenance inspection, video surveillance and in-vehicle monitoring systems will be explored more in detail.

2.2. Video surveillance system

Video surveillance cameras are everywhere nowadays, helping humanity against crime and maintaining close observation on anything from anywhere. Video surveillance is a system consisting of one or more cameras connected to a control system to communicate with each other [6]. A video surveillance system is used to monitor the behavior or activities of certain entities from a distance. A typical video surveillance system has multiple cameras which record continuous videos, a control unit to control the behavior of the camera, some storage devices to store the recorded video, and graphical user interfaces to set up the camera and settings, search and play videos anytime, live view the videos, etc. [6]. Scopes of a video surveillance system include prevention, reaction, detection, and intervention depending on the places and fields it is used. Both analog and digital cameras are used in a video surveillance system. For tracking, recording, and video analysis, an analog camera transmits video through a coaxial cable, while a digital camera transmits video via a network to a single central location [7]. While the analog camera has a low initial cost, widespread compatibility, and very good camera performance under the difficult scenario, the digital camera system provides better image quality and easy remote access. Also, it is easy to integrate and is cost-effective [7]. Many companies in the market provide hardware and software solutions for monitoring in many different scenarios using the video surveillance camera system. Some of the most common use cases are office, house, roads, highway monitoring.

2.2.1. Mobile surveillance camera

The mobile surveillance camera can be a mobile-head camera that can move and tilt [8] or be a fixed camera mounted to a moving object. Kitahara et al. [9] introduced research to develop a surveillance camera network, 'ViSE,' which uses environmental and mobile cameras to address these problems. The system uses a semi-automatic monitoring approach where mobile cameras with larger viewports are used to minimize blind spots and make monitoring work easier. Cameras are calibrated with one another; hence it is easier to track down any moving objects seen at a particular location before. Using mobile cameras raises some privacy concerns in specific places. Researchers used stealth vision with a mobile camera and faded out the privacy regions to maintain privacy [9]. Jackson et al. [10] represented a mobile video camera system that takes aerial video data of roadway locations for vehicle safety analysis. The system records and stores user's behavior on the road, such as speed and lane change, and analyses it to better understand the driver's behaviors on the road. Moreover, this system helps identify accident-prone areas on the road. Greenhil et al. [11] proposed a system where they implement virtual observers in a mobile surveillance system. They used low temporal panoramic video cameras mounted on the buses.

Mobile video recorder systems have been used in such operations where direct human intervention is impossible. Chai and Wang designed and implemented such a mobile video surveillance car which can be used in narrow and unsafe spaces to rescue something. They showed that their system reduces transmission delay, video images are smooth, and streaming is stable [12]. The system uses a USB camera to collect video information, which reduces the CPU occupancy rate, and the video is more fluid.

In summary, most mobile surveillance systems use a single fixed camera mounted on a moving object. These systems mainly vary on the moving object which carries them. A system with one fixed camera is easy to install and maintain but can cover only a limited space. On the other hand, multiple camera systems can cover a wider space but are costly and complex. With the increase in the number of surveillance cameras, problems related to multiple camera systems are increasing, such as operational hazards and privacy concerns [9]. Mobile surveillance cameras help to overcome many of these problems.

2.2.2. Smartphone as a surveillance camera system

Though there have been many important efforts, image and video data collection for the surveillance system is still a complicated, costly, and time-consuming process. In many studies, video data is collected from fixed locations using the pre-set camera. However, the problem arises when video cameras do not exist at the locations of interest [12].

With the advancement in mobile technologies, people are expecting more and more from their mobile devices [13]. As smartphones are becoming powerful, many surveillance systems are making smartphones a part of the system. For instance, Smartphones can be used as a surveillance camera or the processor of the system. Ma and Li [13] designed an intelligent moving vehicle equipped with an Android phone and a USB camera. They developed an Android application to control a vehicle remotely that has a camera mounted in front. Users can specify the direction and path for the vehicle and camera with the Android application. The system is equipped with a Wi-Fi module that takes care of communication between the Android client and the camera module. Because of hardware limitations, using the smartphone's camera as a full surveillance system is not common. Moreover, many systems nowadays use mobile phones as terminals to provide real-time information. Pang et al. [14] proposed a system where a processor connected with a USB camera serves as the server and an Android smartphone serves as the client. The server and client are connected through a socket connection. The client receives video data and sensor information and takes necessary action according to the sensor information. Abdelbacet et al. [15] proposed a similar system for a smart city that uses HTTP protocol instead of socket connection. It uses a three-tier architecture where a separate server is used to process the sensor data. An Android client is used to retrieve the processed data from the server using the HTTP protocol and shows it through a web application. This allows the smart city users to observe the video irrespective of time and place. While most of these systems use a dedicated camera, Tangtisanon et al. [16] designed a surveillance car that uses a smartphone as the surveillance camera. The system consists of two Android mobile phones that have the same application installed. Both systems are connected through a common network. One phone is used as a controller to control the wheel of the car. The other is the surveillance camera which is attached to the car. Video recorded by the second phone is sent to the first phone and shown in the application UI.

2.3. In-vehicle monitoring system (IVMS)

The in-vehicle monitoring system is a collection of electronic devices installed in a vehicle to collect vehicle data, such as location and mechanical information. IVMS can store data for quick retrieval or transmit it to a remote server. Uses of IVMS can vary from device to device. Typical uses of IVMS are recording live data on speeding, acceleration, braking, seat belt use, etc.

2.3.1. Uses of IVMS for road surface monitoring

Lots of IVMS have been developed to improve driver visibility and detect hindrances on the road. Such systems record videos and analyze them instantly to give some insight into the surroundings to the drivers and do not store the videos for later use. Sohn et al. [17] developed a monitoring system for mobile navigation devices based on HD videos recorded by three vehicle-mounted cameras. Two of the cameras were mounted on both side mirrors, while the other was mounted on the rear window. These three cameras stream live videos of blind spots to the screen in front of the driver. They did not use a camera in front of the vehicle as that's already the most visible part to the driver. Okamoto et al. [18] patented a similar vehicle monitoring system where they used a pair of cameras to monitor the surroundings and a distant place from the vehicle. The camera helps the driver see the vehicle's surroundings and blind spots from his driving seat. They used twin cameras mounted in the vehicle between 50 to 90 degrees, ensuring that at least portions of the ranges to be captured overlap with each other.

One of the wide uses of IVMS is monitoring road surfaces. Not just private companies are interested in monitoring road surfaces for their purposes, governments have been spending a great deal to provide safer road networks by knowing the road surface condition in many ways. The most used way is getting the report from the drivers, which is a slow process and does not give the update of all possible road damages [19]. To overcome these problems, Jinwoo Jang et al. [19] proposed an innovative system that automates collecting the road condition data. The system uses multiple vehicle-mounted sensors like accelerometer, GPS, and microcomputer to collect location data and driver's behaviors in certain positions. A supervised machine learning algorithm is then used to identify common patterns in driving by different vehicles in certain coordinates that identify possible hazards on the road surfaces. With some tests in real roads condition, they showed that their system is more efficient than the traditional way.

When it is about collecting road surface data, cameras and sensors in a moving vehicle can provide sufficient data. Some researchers proposed systems to detect the cracks for maintenance. Quintana et al. [20] presented a computer vision system that can detect and classify cracks on the road by a single mounted camera in a truck. The analysis is done in two steps because of the hardware limitation in a moving truck. In the first step, images of the road are collected with geo-location during the daytime. In step two, during nighttime, the main analysis is done, which includes Region of Interest (ROI) delimitation, crack detection, and classification. Hough Transformation (HT) is used to detect lines, and then cracks are composed of their composition. Cracks are classified based on spatial properties. Sy et al. [21] proposed a similar system to detect road cracks in France. The system uses two-line scanner cameras, and two laser illuminators mounted in a vehicle to illuminate the road surface. The system works under any light conditions because of these external light sources. Defect detection on the surface is mainly done with manual checking by an operator. Histogram equalization and morphological operation are used to filter the image without a crack to reduce the number of images to be checked. They proved that their system is fast and efficient in filtering a no-crack image in real-time.

2.3.2. Smartphone as IVMS

IVMS needs various sensors such as GPS, Gyroscope, Speedometer, Accelerometer, Magnetometer, etc. Installing dedicated sensors in vehicles or roadside is expensive [22] [23]. Furthermore, installing these sensors in many vehicles is time-consuming and requires more resources and effort [24]. As a result, smartphones are widely used to implement IVMS as they are equipped with various built-in sensors. For example, smartphones are used in road surface monitoring systems where smartphone sensors collect location and surface images [24] [25]. Chien et al. [26] developed an Android-based warning system for lane detection and departure warning. The system uses an Android camera to capture the image of the road, and the multi-core CPU of the Android platform is then used to process the image and make the decisions. The system uses an adaptive threshold algorithm which is proven to be more efficient than the existing lane detection systems. Vittorio et al. [25] developed a similar system where they used an Android application with GPS and accelerometer to get acceleration data of the vehicle. With a statistical analysis of collected data, they decide on road conditions to make the road safe and comfortable. Taniguchi et al. [27] used bicycle-mounted ultrasonic sensors to identify surface conditions and notify riders immediately. Their Arduino-based system can detect the surface fault of 223 cm away from a moving bicycle.

One of the most important uses of IVMS is to improve driver and passenger security by providing surrounding data of the vehicle. Jheng et al. [28] proposed a system for forwarding vehicle detection and collision warning using an Android smartphone application. The android camera takes a picture of the road, and then the Android mobile CPU processes the image to detect vehicles and threats by using the Bayes classifier. On most sunny days, the system has shown more than 92% accuracy. Like the vehicle's surroundings, the driver's behavior is also important for safety on the road. Manoharan et al. [29] present a novel approach of detecting eye rub due to irritation in the eye and yawning detection through the intensity sum of the facial region. They presented a system that uses Android OpenCV in a low-cost Android device to identify facial expressions.

Smartphones can be used to crowdsource road condition information when needed. For example, drivers on the road can report the condition of the road surface to a remote system. Ghose et al. [30] proposed such an application for in-vehicle smartphones where users can report any disturbance on the road. The report is then shown in a web application to the other users. The application can also be used by city authorities who can plan repair works using the data.

The use of the smartphone in IVMS is enormous. With the advancement in smartphone hardware and software, many other different uses of IVMS can be revealed by smartphone technology.

2.4. Software Testing

Software testing is a series of investigations conducted on software or service to ensure quality [31]. The main objective of software testing is to identify software bugs so that they can be corrected. However, It can also provide objective, independent information about the quality of software and the consequences of its failure to developers, users, and sponsors. Software testing is done under certain conditions and cannot always ensure that the product will function the same under all conditions.

A very effective way of conducting software tests is separating "development" and "validation". During the development phase, a black-box and white-box testing approach can be used to test the application's functionality. Black-box testing is done throughout both the software development life cycle and software testing life cycle [32]. The advantage of black-box testing is that the testers and developers can work independently. Testers do not need any technical knowledge about the software, and testing is done from the user's point of view [33]. Black-box testing is easy to execute and gives a quick overview of the functional readiness of the system by quick spot check [35]. For white-box testing, testers need to know the internal logic, structure, and full knowledge of the source code [32]. It improves code efficiency. It also helps to remove unused code in the software.

The software needs to be tested with experts and intended users. In addition to the participants' principles of safety and data protection, all the tests and experiments with users should fulfill the principles of objectivity, reliability, and validity [35]. Testing with intended users is done with the people who are not involved in the development process. They can be non-expert drivers when testing automotive software; hence it needs more safety measures. On the other hand, testing with experts is done with those who know how the software works and are involved during the development process. It is important to define the scenario before starting a test. Also, target test users should be defined.

Modern software engineering is a human-driven action. From the very early stage of requirement analysis to testing and maintenance, inputs and outputs are tested by humans [37]. Though many parts of software development are automated using advanced tools, the major performers are still human, such as developers, testers, and consumers. Hence the testing and evaluation of software using different human-computer interaction techniques are very crucial to test the usability of the developed software. Evaluation techniques used in HCI can be classified into two types: predictive models and techniques and experimental techniques [36]. Predictive models are theory-based such as System Usability Scale (SUS), Cognitive Complexity Theory (CCT). Predictive evaluation techniques may also be based on HCI heuristics. The user looks for different user interface components that he or she already knows from the previous experience [36]. The most popular predictive technique is the SUS, a 'quick and dirty' way to assess the usability of a product using a survey scale [38]. There are many other similar survey techniques in the industry, such as the After-Scenario Questionnaire (ASQ), Computer System Usability Questionnaire (CSUQ), Software Usability Measurement Inventory (SUMI), etc. The main benefit of these predictive models and techniques is that it helps to evaluate user interfaces even at the design phase and helps to improve the design before implementation is done. It can be done iteratively throughout the development life cycle.

Experimental techniques use real data from real users by doing real tasks [36]. User behavior during experimental studies needed to be recorded for later analysis. Paper scenarios, mock-ups, computer system prototypes, or Wizard of Oz (WOz) platforms are used to do the experiments [50]. The Wizard of Oz testing method is a process that allows test users to test an interface without knowing that the responses are generated forcefully by a human rather than a computer system. WOz can be done throughout the design process as the system evolves [39].

Like any other software application, an in-vehicle software application can be tested in different stages, starting from feasibility study to final release. In their book "A Handbook of Driver Assistance Systems" [35], Winner et al. discussed different driver assistance systems testing stages. For example, a taxonomy based on the v-model [40] of product development is introduced to test the software requirement in the design process. System properties to be tested first decrease down to the bottom of the v-model and increase again to the top. Additional tests are needed, along with product development tests for legislation and consumer protection. Regulators and consumer organizations often set the requirements for these tests.

Testing automotive software can be a real challenge as sometimes it is hard to simulate a real road condition. For example, testing software developed for car airbags needs to simulate a real crash situation to get the expected test results [41]. Sensors like Accelerometer, Speedometer, and GPS, which are widely used in IVMS, are hard to simulate in indoor development premises. Moreover, many countries have restrictions on using mobile devices while driving and need special permission from the authorities to run tests on the road. For example, in Finland, using mobile phones without a hands-free device is prohibited (Road Traffic Act of 1 January 2003) [42]. As a result, direct interaction with the software is not possible while driving. ISO-26262 [43], published in 2011, defines the safety standard for testing automotive software systems addressing the need for an international safety standard specific to safety-critical automotive applications. ISO-26262 is intended to be applied in electrical and/or electronic systems of vehicles related to passengers' safety in a passenger car with a maximum gross mass of 3500 kg.

3. ROUTA SYSTEMS

Routa Systems[3] is a collection of applications developed by Ficonic Solutions Limited. The main purpose of Routa Systems is to manage employees and road work, like handle work inspections and manage employee work time. Currently, Routa Systems support three languages which are English, Finnish, and French. Routa Systems have different software modules. In this chapter, Routa mobile application, the Road recorder application, and the Routa web application are described, which are most relevant to this thesis. In addition to these, Routa Systems have some other applications and variants used for company-specific use cases. For example, there is a time tracker application for the workers to help them track their work time and to create their allowance report. Also, the Routa Android application has a different version named Routa Forest, designed and built to work in the forest to track worker's location and status and manage work time.

3.1. Routa Android application

Routa Android application is a mobile application developed for road construction work management. Routa Android application is responsible for various tasks, such as creating road observations, work reports, and vehicle reports. Routa Android application is compatible with Android SDK version 23 (Marshmallow) and higher. The application has a login interface that authenticates users to use the application. After the login, users see three tabs: the 'WORK' tab to manage work, the 'OBSERVATION' tab to create a road condition report, and the 'MAP' tab to see the map view with real-time vehicle position. On the work tab, users can select which vehicle they are using while working on the road. Also, users can specify in which customer, contract, and the task they are working on. Moreover, users can indicate if they are working on the task (work mode in the application) or are ready to move for the next task (resettle mode in the application). When the user is in work mode, the trip log and work history are saved and synced with the remote Routa Systems backend server. When the user is in resettle mode, their location and work status is not saved in the remote backend, and the application will be reset and ready to move to the next task. Figures 3.1 present the WORK tab view of the Routa Android application.

The live map view, presented in Figure 3.2, allows users to see their current location. The road information is shown at the top bar of the application. The live map view also facilitates search features; for instance, users can search a road or bridge with a road number or bridge number.

Routa Android application can also be used to report vehicle inspection where the condition of vehicle parts is reported for further action (see figure 3.3). Routa Android application is also used as the controller of the Road Recorder application. It provides the necessary authentication credentials to the Road Recorder application and triggers different events for it, depending on the user's work mode and other metadata. Logging out from the Routa Android application also logs out users from the Road Recorder application.

[3] https://routasystems.com

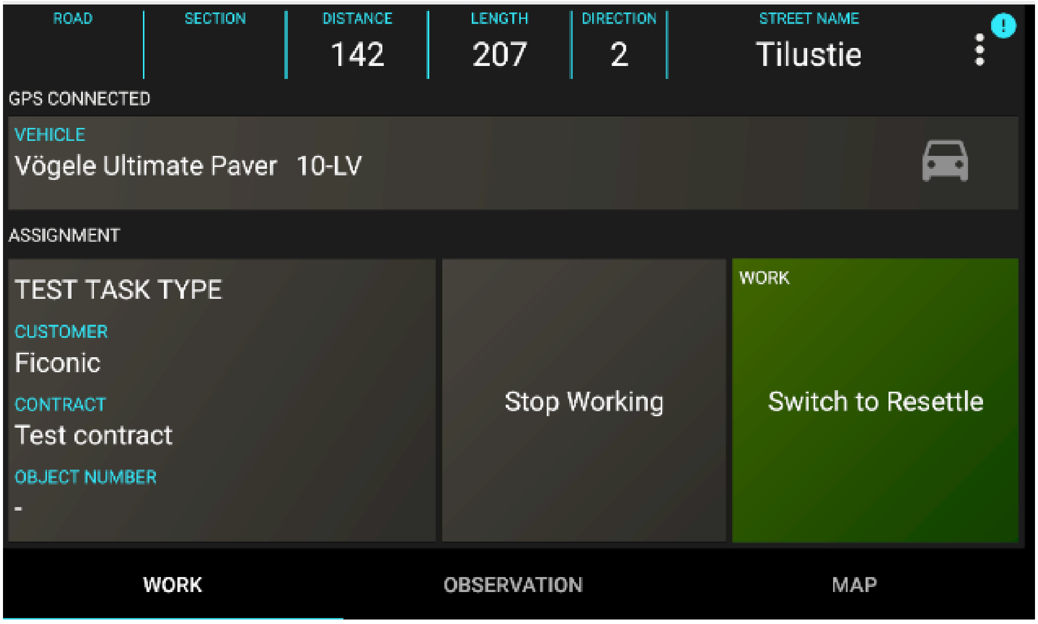

Figure 3.1. Work view of Routa Android application.

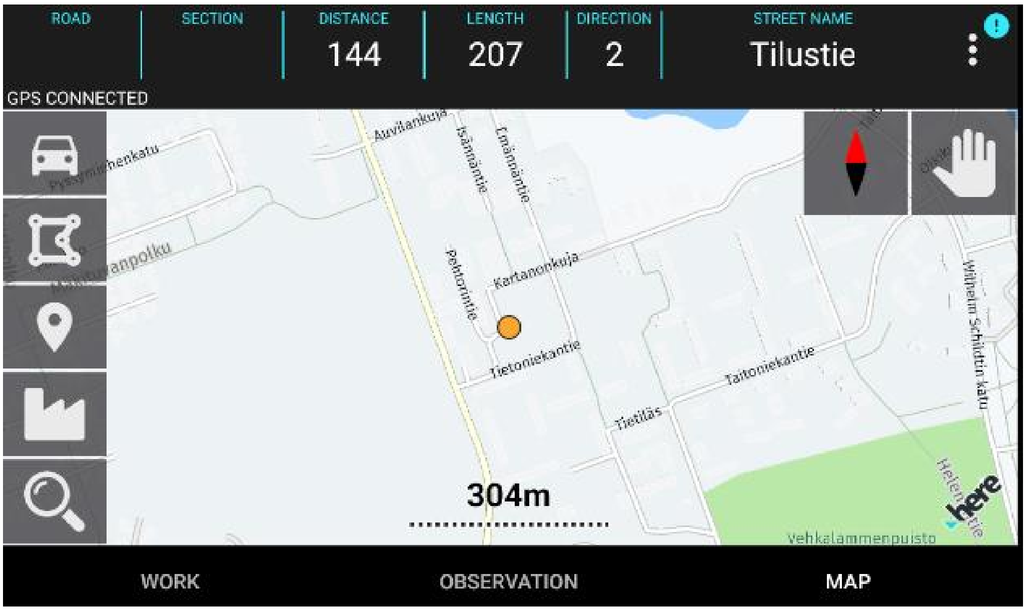

Figure 3.2. Live map view of the Routa Android application.

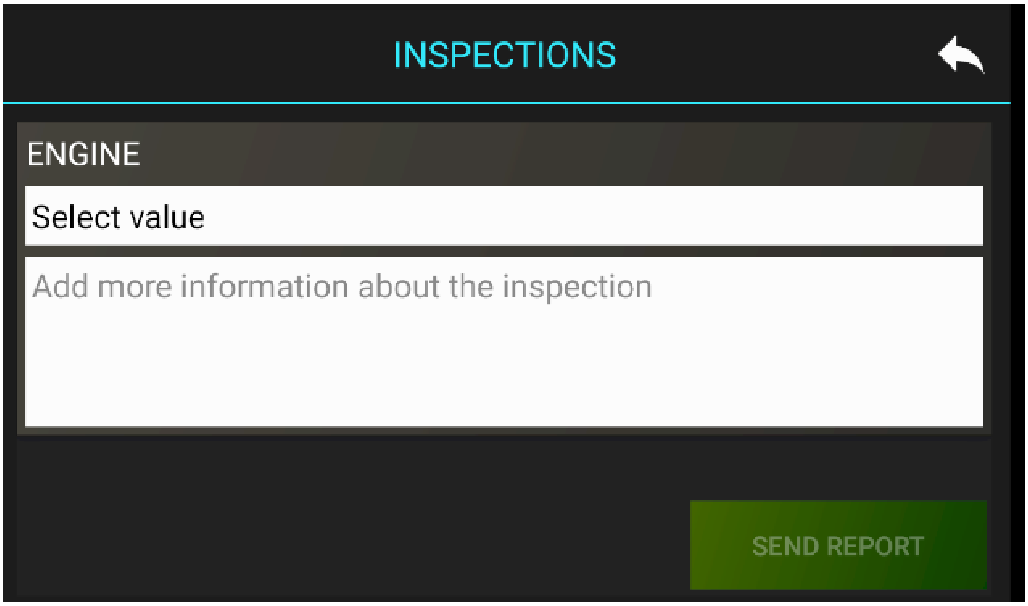

Figure 3.3. Vehicle inspection view.

3.2. Road Recorder application

The Road Recorder application is the video recorder application that records the road surface's live video stream and geographical coordinates. The development and evaluation of the Road Recorder is the focus of this thesis. This application is tightly coupled with the Routa Android application and cannot be used without the Routa Android application running. The Road Recorder application uses the credentials from the Routa Android application to authenticate users. The application's main purpose is to record the road surface video using the Android device's built-in camera and collect geo-location data of those videos. The functionality of the Road Recorder application and its user interface will be discussed in more detail in chapter 4.

3.3. Routa web application

Routa web application is a task management browser application for managing the company's employees and road work. This application is compatible with any standard browser. The application provides a login console where users with different roles such as driver, map viewer, supervisor, and admin can log in to the system (see figure 3.4).

The application has many features that help road work and construction companies with their contracts and tasks. Ficonic solutions have been adding new features now and then to the application upon the customers' needs. The most relevant functionalities of this thesis are described in Sections 3.3.1 and 3.3.2.

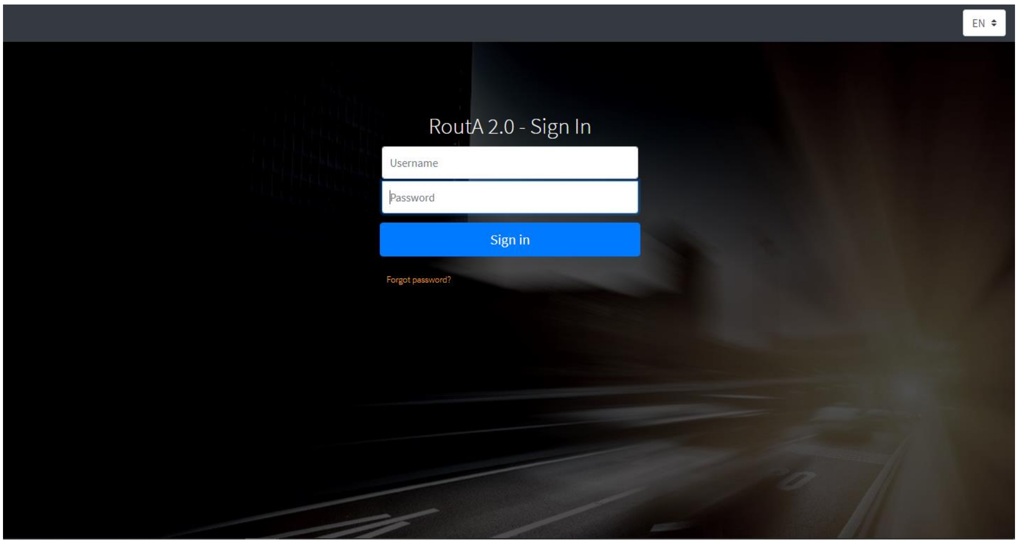

Figure 3.4. Login view of Routa web application.

3.3.1. Work management application

When users first log in to the web application, they see the menu on the left of the window, where each item contains sub-menus. The main menu items are Map, Trips, Observations, Areas, Work safety, Supervisor, Fleet, Settings, and Admin. The menu varies depending on user rights. For example, workers, supervisors, and admin users have different menus. Map item, which opens a map view, is selected by default (see figure 3.5). On the map, the user can filter and see different components depending on user rights. The map shows the last registered vehicle's position, area of work, road observations, etc.

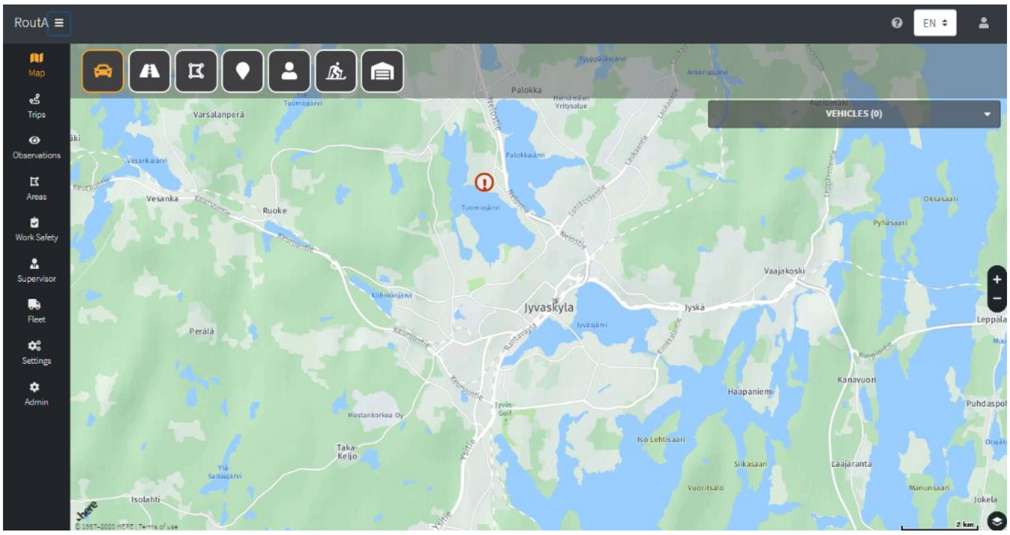

Figure 3.5. The default landing page with menus of the web application.

With the Admin item sub-menu, admin users can do most of the administrative tasks such as adding users and editing their access rights, handling customers and contractors' relationships, etc. The supervisor item sub-menu is for supervisor users responsible for managing entities related to work, such as handling contracts and orders, generating work reports, creating and assigning work assignments to the worker, etc. Different application settings, such as predefined task types, email lists, etc. are managed with the Settings item sub-menu. Area item sub-menu allows managing the work area. Fleet item sub-menu is used for work vehicle management, Observation item sub-menu to manage observations such as road damages, and Trip item sub-menu to browse road trips of the vehicles. The latest addition in the application is the Work Safety item, which allows the worksite and worker's safety. Supervisor users can see from this item if any user alert or danger report has been submitted by worker users using the Routa Android application and take necessary actions.

A demo version of the application[4] is available and can be accessed with a username and password on request.

3.3.2. Road Recorder management application

Road Recorder management application, which is part of this thesis, is planned to be a part of the Routa web application. It will be a separate sub-menu under the Observation item. In the Road Recorder management application, authorized users, such as the supervisor users, can browse the videos and location metadata that are uploaded by the Road Recorder application. Section 4.1 and 4.3 of chapter 4 will discuss the functionality and user interface of the Road Recorder management application.

[4] https://app.demo.routaapi.com

4. DESIGN

This chapter will first discuss the requirements for Road Recorder and Road Recorder management applications. Then it will briefly explain the system architecture, followed by the description of system functionality. Finally, the user interface of the system is presented.

4.1. Requirements

The first step of design was defining the requirements with the customer. Requirements are divided into different categories and analyzed to fit them with the feature requests from the customer.

4.1.1. Functional Requirements

Functional requirements are product features that accomplish their tasks. They define a system or system components. Functional requirements describe the functions a software must perform. Functional requirements for Road Recorder applications were defined after discussing different use cases with the customer. The functional requirements of the Road Recorder and Road Recorder management application are listed below.

Road Recorder Android application

1. Wireless connection between host devices of Routa application and Road Recorder application.
2. Unobtrusive connection of the devices.
3. Automatic reconnection of the applications and devices.
4. Start and stop recording automatically.
5. Record and store geo-location points as metadata of the videos.
6. Upload recorded video and metadata to the server automatically and asynchronously.
7. Track uploaded recordings, recordings in the queue, and upload failed recordings.
8. Watch video recording display preview.
9. Always keep the video focus on the road surface.

Road Recorder management application

10. Search videos with metadata (Address, Work contract, and order, etc.).
11. Select and play front and rear videos side by side to compare road conditions.
12. Ability to watch the video at a certain geo-point.

13. Enable the location tracking on the map while playing a video.
14. Automatic detection of road cracks on the video.
15. Automatic detection of road fixes.

4.1.2. Performance requirements

Performance requirements describe how well certain system components perform their tasks. Poor performance causes a bad user experience.

Response time:

Road Recorder Android application

1. Establishing the connection between Road Recorder and Routa Application should be done within 3-5 seconds.
2. Starting and stopping a recording should be done within 1 to 2 seconds. Otherwise, it can cause missing spots in the videos, hence cause valuable data loss.

Road Recorder management application

3. In the Road Recorder management application, searching and playing videos are demanding tasks for response time. Users should be notified of the progress.
4. Feedback should be given to the users if the task is delayed for more than 5 seconds or there is a failure.

Scalability:

Road Recorder Android application

5. The application should be available for use on appropriate devices at any time and place.
6. Automated testing should be implemented to reduce bugs.

Road Recorder management application

7. The server and database should be scalable.
8. The database should be indexed to perform an efficient video search.

Platform:

9. The Road Recorder application should be compatible with Android 6 Marshmallow (API 23) and higher versions.
10. The Road Recorder management application should be compatible with any modern JavaScript-enabled browser.

4.1.3. User Interface Requirements

User interface requirements define how the graphical layout of the application should be presented and how such components should behave when a user interacts with them. A well-designed interface makes it easy for users to understand the functionality, purpose, and control of a system component. The user interface requirements of Road Recorder and its management system are listed below:

Road Recorder Android application

1. The Road Recorder application user interface should be compatible with all screen resolution sizes, including tablets and large screen mobile phones.
2. It should only support landscape mode.
3. The application should contain common android interface design components to understand each icon's meaning easily.
4. The application should support multiple languages.

Road Recorder management application

5. The Road Recorder management application should be easily resizable in the browser.
6. Interfaces should be mobile browser-friendly, and responsive.
7. Standard error messages should be shown to the users.
8. Localization of the user interfaces.

4.1.4. Security Requirements

Security requirements are a set of targets to protect the application against security vulnerabilities. Security requirements can be set for both functional and non-functional components.

Data Security:

1. User’s passwords should be encrypted before saving in the storage.
2. Only authorized users with proper access rights should see the information of other users.

System Security:

3. Secure communication between the Web servers and other components.
4. Proper security configuration at the server-side protecting from unauthorized access.

4.2. System architecture

The overall system architecture is presented in Figure 4.1.

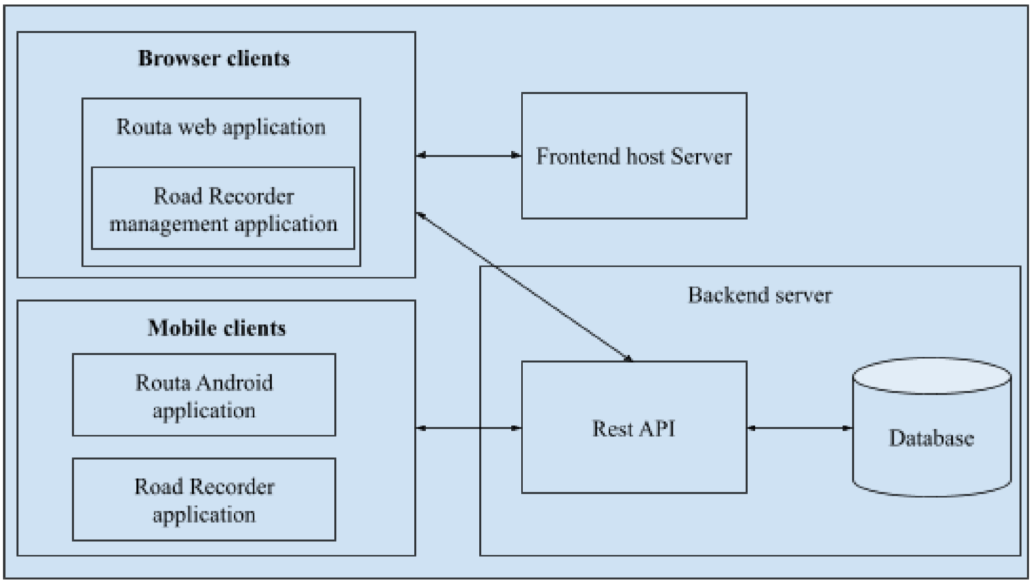

Figure 4.1. System architecture of the Routa system.

As can be seen, the system consists of two servers, one for the backend API and the other for the frontend. The API server hosts the RESTful APIs, which communicate with the database. There are two types of clients, browser clients and Android mobile clients. Routa web application and the Road recorder management application, which is also a part of Routa web application, are the browser clients. The mobile client includes the Routa application and Road Recorder application. UI server hosts the browser clients. API server receives and processes the requests from the clients. If the request is to update or send data, then the API server connects with the database for corresponding actions and notifies the clients back. Client applications show the data to the users.

4.3. Functionality description

In this section, the expected behavior of the systems is presented from both worker and supervisor roles. The worker uses the in-vehicle Road Recorder application, while the supervisor uses the video recording management functionality of the Routa web application, refer to Section 3 for details.

Road Recorder application

Since the Road Recorder is an in-vehicle application and the drivers use it while driving, it shouldn't have a lot of functionality with direct user engagement. Figure 4.2 shows the functionality use case diagram of the Road Recorder application for the worker role, incorporating also functional requirements given in Section 4.1.1.

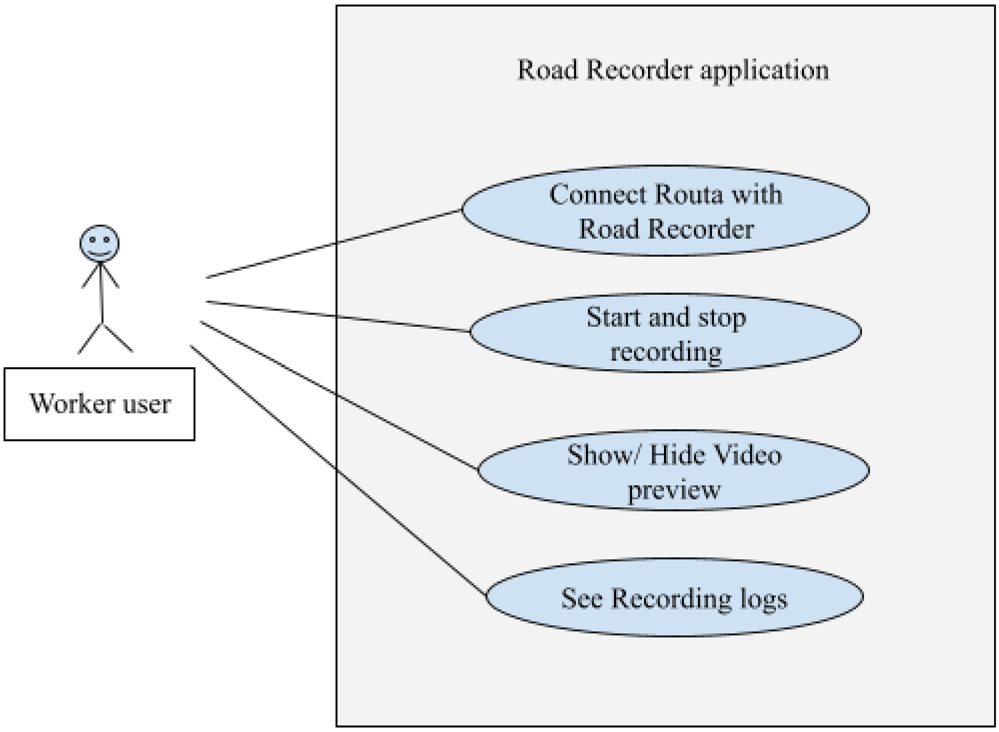

Figure 4.2. Use case diagram of Road Recorder from worker user role.

The worker users who will drive the maintenance vehicle on the road are the intended users of the Road Recorder application. The main functionality of Road Recorder is to start and stop recording. Users will need to log in to the Routa application and connect the Road recorder with the Routa to use the Road Recorder. When both applications are connected, recording video and metadata will start automatically depending on certain conditions, for example, if the Routa application is in work mode and the vehicle is moving at a speed greater than 5 km/h and less than 40/h km. If the vehicle crosses this limit, the recording will stop automatically. The driver of the vehicle will be able to switch between on and off of video recording preview. There will be a recording log menu where users will see uploaded recordings, upload pending recordings, and upload failed recordings logs.

Road Recorder management application

The supervisors are the intended users of the Road Recorder management application. Since Road Recorder management is a part of the Routa web application, the user will log in to the Routa web application to use the recording management tool. Figure 4.3 presents the functionality use case diagram of the Routa Road recorder management tool, incorporating the functional requirements presented in section 4.1.1.

When a supervisor wants to inspect the recording, the first thing he/she will have to do is to search for the appropriate video. The supervisor will be able to use the search tool, enabling filtering the videos using different parameters, such as work order, video recording's city, timestamp, etc. Users will select front and rear videos and play them side by side to compare them. They will also be able to see the geographical location of the place where the video was captured on the map while playing the video in the player. Also, clicking on a specific geo-location will open the video of that point.

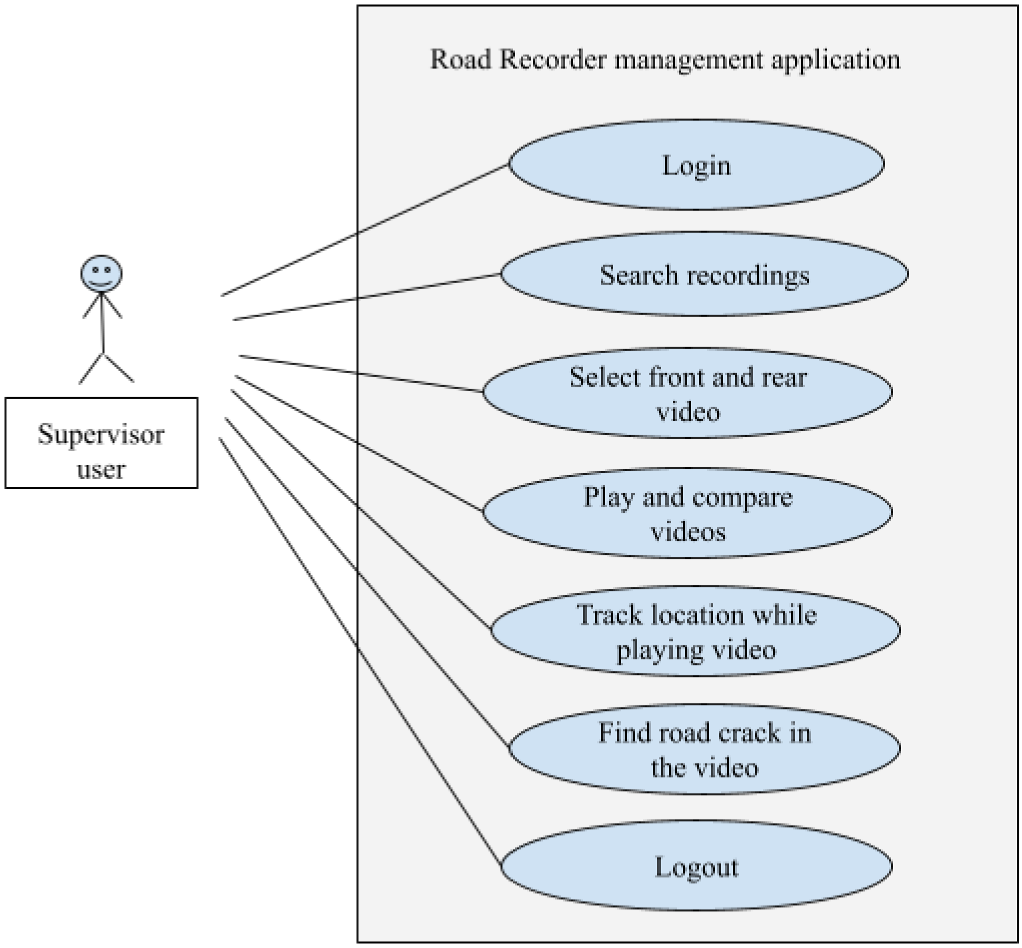

Figure 4.3: Use case diagram of Road Recorder.

4.4. User interface

The UI design process is presented with the diagram in Figure 4.4. As can be seen, from the UI design process, we first identified the requirements for our application. After that, we started brainstorming sessions for starting the UI design process. For the Road Recorder management application, already existing Routa web application design guideline was used. Similar colors, fonts, icons, and themes were used to be consistent. Since the Road Recorder application should not include any distracting features and based on the UI requirements presented in section 4.1.3, we decided to use basic Android material designs. Prototypes for both applications were created directly using Android Studio[5] where the design without any functionality was implemented. Such a solution allowed to make modifications easily. These prototypes were then discussed and tested with some real users at the client site. The design was finalized After a couple of tests, modifications, optimizations, and discussions.

[5] https://developer.android.com/studio

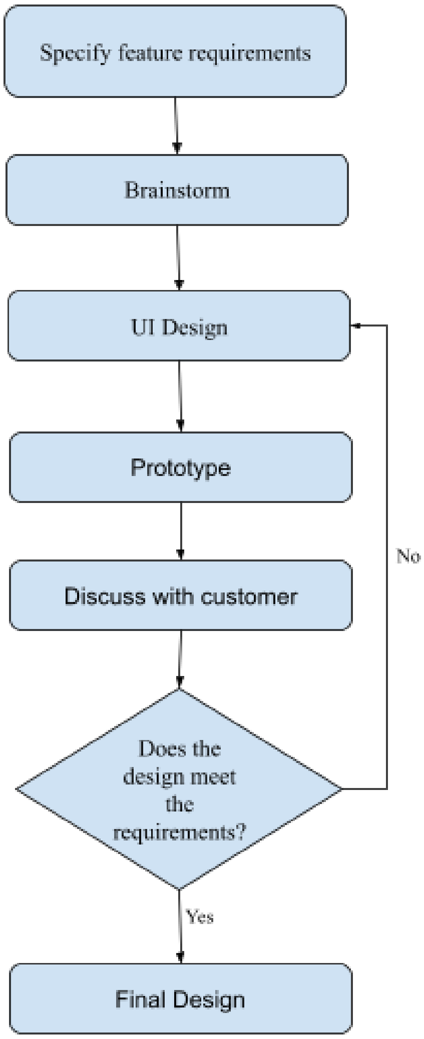

Figure 4.4. Flow diagram of the design process.

5. IMPLEMENTATION

The implementation of Road Recorder Android and management applications had been an iterative process. After analyzing the requirements and getting the design done (see Chapter 4), the actual implementation has started. Each of the features was implemented followed by one or more testing and evaluation phases with the customer. Some of the features were left out for cost-saving reasons. Examples of such features include automatic detection of road cracks and road fixes in the videos. Finally, choosing the technology stack was dictated by already existing Routa Systems applications.

This chapter first discusses the development environment and tools used to implement the Road Recorder Android and management applications. After that, backend implementation will be discussed, including server technologies and third-party services. Then, the implementation of the Road Recorder Android application, followed by the implementation of the Road Recorder management application, is explained. Finally, the user authentication process of the system will be discussed briefly.

5.1. Environment and tools

Windows 10 was used as the operating system to develop both Road Recorder Android and management applications. The tools and services that were used to develop the applications are briefly described below.

GitHub

GitHub[6] is a web-based GIT repository hosting service. It offers the distributed version control and source code management (SCM) functionality of GIT[7], as well as adding its features. The source code of developed applications was stored in the private repositories of GitHub.

Android Studio

For the development of the Road Recorder Android application, the Android Studio[8], a free Integrated Development Environment (IDE) distributed by Google was used. It is an all-in-one tool to develop professional Android applications. Android Studio comes with the necessary software development kit (SDK) to download the facility for all Android versions. Android Studio version 3.0 or higher was used for the development. Android Studio also offers an extension for GitHub integration, which takes care of most of the settings.

---

[6] https://github.com

[7] https://git-scm.com

[8] https://developer.android.com/studio

WebStorm

For the development of the Road Recorder management application, the licensed version of WebStorm[9], a popular Web development IDE developed distributed by JetBrains[10] was used. It supports almost all the popular Web programming languages, including but not limited to HTML, CSS, JavaScript, etc. It also comes with a nice GUI to manage GIT repositories.

PhpStorm

A licensed version of PhpStorm[11] was used for the development of the backend of Road Recorder applications which is also a commercial product of JetBrains. Just like WebStorm, it also has similar extensions for GIT integration.

Other tools

In addition, some other tools were used. For example, to browse the database MySQL Workbench[12] version 8.0 was utilized, and Postman[13] for testing the REST API.

5.2. Backend implementation

The backend is the server-side of the application. It stores and manages data and supports client-side applications. The backend is not visible to users. The backend consists of the Rest API, database, and server, see Figure 4.1.

5.2.1. Server implementation

For the development of the PHP backend, the built-in Web server[14] available in PHP 5.4.0 was used. For the production deployment, the Apache webserver[15] was used. Amazon Web Services[16] are used as the cloud platform for the backend and Web frontend.

Rest API is written in PHP (version 7), which is a server-side scripting language. The Symfony17 framework (version 4) was used. It is a modern PHP framework, providing easy and compact functionality that speeds up the creation and maintenance of Web applications.

---

[9] https://www.jetbrains.com/webstorm
[10] https://www.jetbrains.com
[11] https://www.jetbrains.com/phpstorm
[12] https://www.mysql.com/products/workbench
[13] https://www.postman.com
[14] https://www.php.net/manual/en/features.commandline.webserver.php
[15] https://httpd.apache.org
[16] https://aws.amazon.com
[17] https://symfony.com/releases/4.0

5.2.2. Database

The database stores information related to the user, job, order, vehicle, location, recordings, etc. MySQL[18] database is used as the data storage. MySQL was selected because it is free to use and has extensive support for working with spatial information. We used Doctrine[19], an open-source PHP object-relational mapper library for writing SQL queries.

Most of the data received from client applications are in JSON format. JSON data is then converted in the backend to the format and structure needed to store it in the database.

Since this thesis focused on designing and implementing the Road Recorder and the Road Recorder management applications, the main unit of information to store in the database was recorded metadata. Therefore, a dedicated table 'Recordings' was created in the database. Its schema is presented in Table 5.1. Recorded video files are stored in Amazon S3, and the link to the file is put in the database.

Table 5.1. ‘RECORDINGS’ table schema

Column Name	Data Type	Description
id	int (11)	Unique Id of the record.
videoFile	varchar (255)	Link to the recorded video file.
startTime	datetime	The start time of the recording.
endTime	datetime	The end time of the recording.
geometry	longtext	Road geometry (Latitude and longitude)
workOrder	int (11)	Foreign key to a Job Id.
vehicle	int (11)	Foreign key to the vehicle which is used to create the recording.

The Recordings table is integrated into the rest of the database developed for Routa Systems. Figure 5.1 shows the ER diagram of the part of the Routa Systems database that is connected with the Recordings table. As can be seen, along with other metadata, the Recordings table stores the job id (workOrder) and vehicle id related to a recording. This information is stored as foreign keys to Orders and Vehicle tables respectively.

[18] https://www.mysql.com
[19] https://www.doctrine-project.org

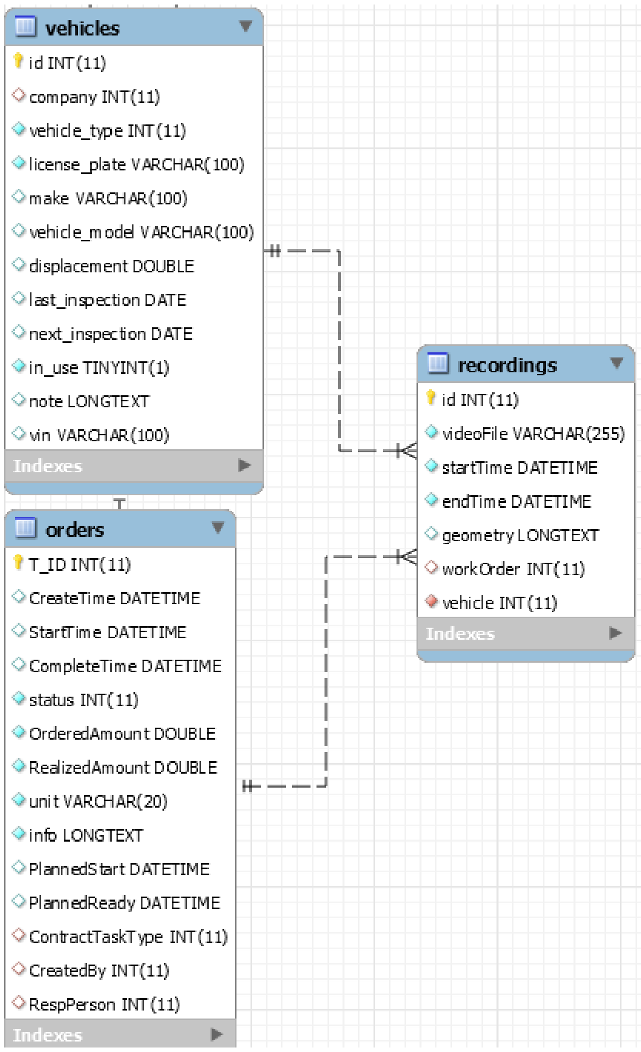

Figure 5.1. ER diagram of the Routa Systems database.

5.2.3. Implementation of the Authentication process

Routa Systems authenticate users using a username and password. Passwords saved in the database are encrypted using the SHA256[20] cryptographic hash function after adding salt. Every time a user makes a service call, he/she is authenticated to the server using a JWT[21] based token. This token is supplied to the clients when they successfully logged into the system using a username and password. The token is generated using the SHA256 algorithm, where a single secret key is used to encrypt and decrypt the token. This token carries the necessary information to authenticate users in each REST API call.

As all the clients of Routa Systems use the same backend, the backend authentication process is also the same for all of them. Road Recorder does not have a login user interface of its own. It gets the necessary authentication token from the

[20]https://www.researchgate.net/file.PostFileLoader.html?id=534b393ad3df3e04508b45ad&assetKey=AS%3A273514844622849%401442222429260

[21] https://jwt.io

Routa Android application every time a recording is completed. This token is then used to send the related recording to the server.

5.2.4. Third-party frameworks and services

Amazon S3[22] is used to store video files. Amazon S3 is a storage service offered by Amazon. It provides SDK for PHP for easy integration, which has comprehensive documentation. Doctrine is another third-party library used as the ORM for MySQL mentioned earlier.

5.3. Road Recorder Android application

Road Recorder application is an Android application for the driver user. It is the data generator of the system. The Road Recorder application is developed with Java (version 8) using the native Android (API 27) framework. The class diagram of the Road Recorder application is presented below.

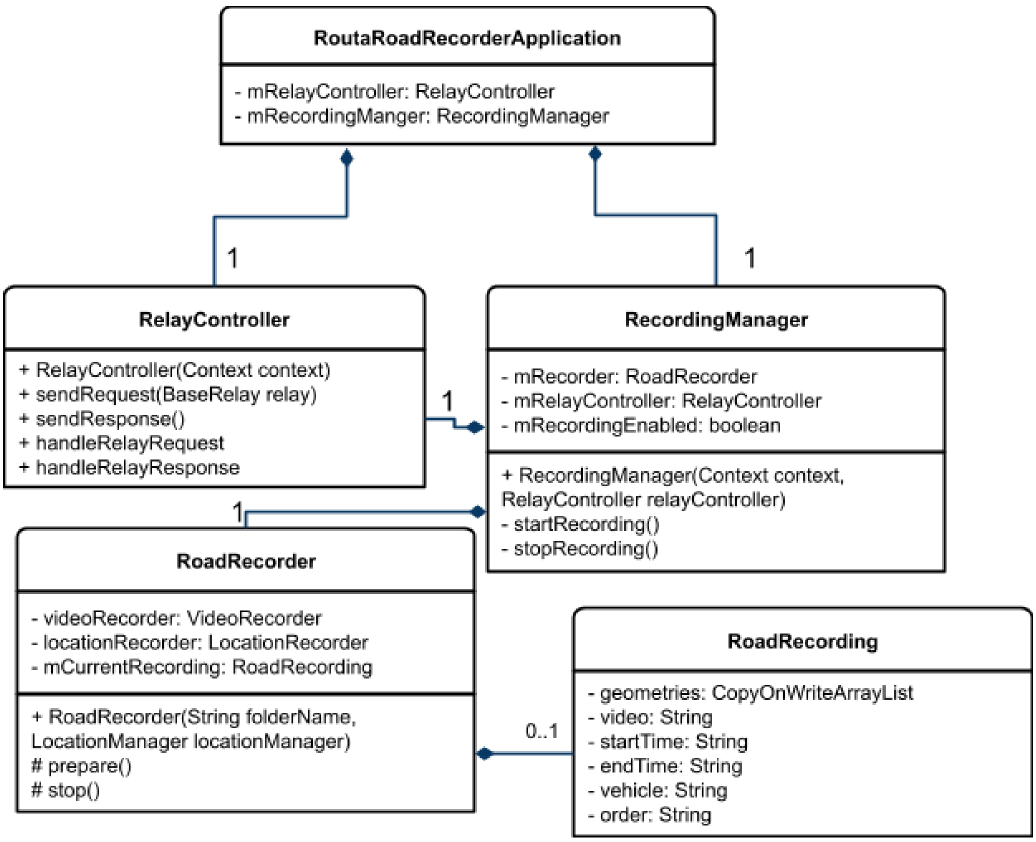

Figure 5.2. Class diagram of Road Recorder application.

[22] https://aws.amazon.com/s3/

The application consists of an instance of RelayController and RecordingManager. RelayController handles the communication between the Routa Android application and the Road Recorder application using P2P Wi-Fi direct. It sends the request and response to the Routa Android application and handles those coming from the Routa Android application. RecordingManager listens to the request from RelayController and starts and stops recording when requested. RecordingManager also notifies recording status to RelayController. RecordingManager has an instance of Road Recorder. Road Recorder is controlled by RecordingManager and records the video and location data. RoadRecording class holds the current recording information before sending it to the remote server.

GUI of the Road Recorder application consists of two simple views. When users launch the Road Recorder application, they will see the home interface, presented in figure 5.3. Here, recording and connection status are shown in the center of the screen and the notification bar. The home interface has two buttons placed at the top right corner of the screen. The first button is to turn on and off the video preview. If the user switches to video preview, he/she can see what the recorder is exactly recording. The second button is to access the recording logs.

Figure 5.3. Home user interface.

The recording log view, which is shown in figure 5.4, manages the recent recordings. It has three buttons: to see the list of recordings yet uploaded, uploaded recordings, and the upload failed recordings, respectively. When a user clicks on one of them, a list of recordings with the timestamp will be shown.

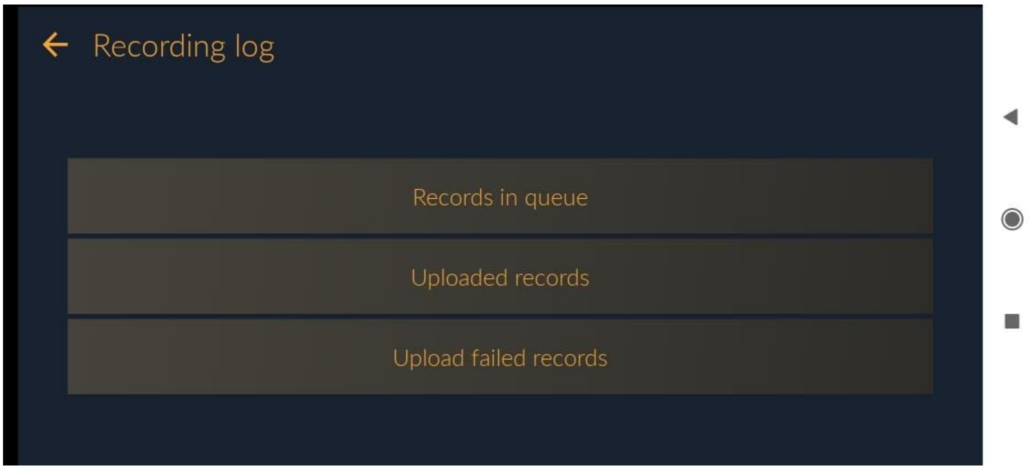

Figure 5.4. Recording log menu interface.

5.3.1. Synchronization with Routa Android application

Road Recorder application uses Android P2P API[23] to connect and communicate with the Routa Android application. When the Road Recorder application is opened and connected to a wireless connection, it first looks for the device running the Routa Android application, which is connected to the same network. Then it connects with the device, starts data and command exchange (see Figure 5.5). The connection uses WPA2 encryption[24] to secure the data exchange. The Road Recorder application accepts commands from the Routa Android application and executes them, and sends a confirmation message of the execution. Depending on the confirmation message, the Routa Android application gets ready to send the next message or packet of data to the Road Recorder**.**

[23] https://developer.android.com/training/connect-devices-wirelessly/wifi-direct
[24] https://www.webopedia.com/TERM/W/WPA2.html

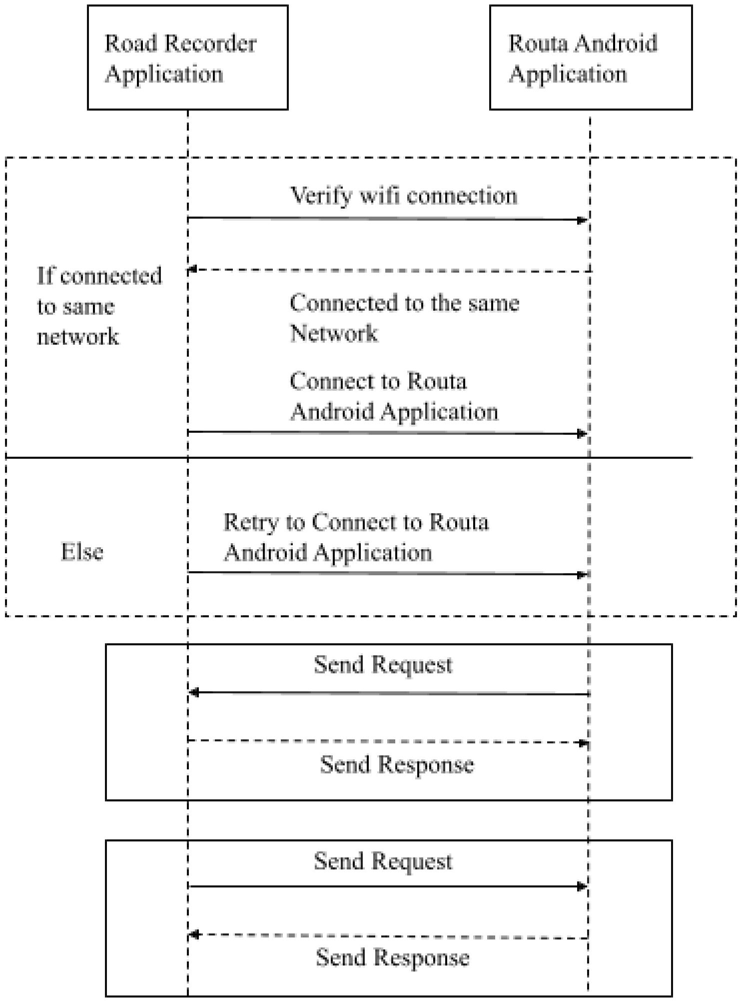

Figure 5.5. Sequence diagram of the connection between Road Recorder and Routa Android Application.

5.3.2. Video recording with Camera API2

The Road Recorder application records videos and metadata of the road. Both the Road Recorder and Routa Android applications need to be connected to the same Wi-Fi connection, and they should be connected. The driving mode in the Routa Android application needs to be in Work mode, and the car should be at a speed slower than 40 km/h and faster than 5 km/h. If all the conditions are fulfilled, the Routa Android application sends a request to the Road Recorder application to start recording. The Road Recorder application starts video recording if the phone's camera is available at that moment and sends a confirmation back to the Routa Android application. We used Android's latest camera API[25] 2. Camera API2 has the functionality to help keep the camera focused on the distant road surface, which is very useful, especially when there is any insect or raindrop on the windshield. Figure 5.6 represents a recording UI while running on Camera API2

[25] https://developer.android.com/reference/android/hardware/camera2/package-summary

Figure 5.6. Recording UI of Road Recorder application.

5.3.3. Collection of location data with geo-location API

When the Road Recorder gets the signal to start a recording, it also records geo-locations of the car with the timestamp. These geo locations are later used to draw the driven road section on the map. These locations are synchronized with the video in a way that when a video is played, a line string will point to the exact location on the map of which video recording is playing at that moment. This synchronization between map location and the video player's position is done by calculating the time difference between the log time of the video starting point and the log time of the selected point. To collect location data, we utilized Android's built-in GPS and Android's geo-location API. Location is recorded and saved in a maximum interval of 500 milliseconds. For the development purpose, the Lockito[26] location simulator was used which helped to simulate location and speed indoor.

5.3.4. Synchronization with Routa backend

When video recording is in progress, timestamp and geo-location are saved in the application's local storage at each geo-location change. Each video can be a maximum of two minutes long to keep the file size small enough for a quick upload. Videos are saved in MP4 format without audio. After two minutes, a new video recording starts. When a video recording is finished, the Road Recorder application tries to send the video and the metadata to the backend asynchronously. If uploading fails due to a network or server error, the recording goes to the queue and retries later. If the upload is successful, uploaded metadata is added to the log, which is then visible from the Recording log view. To upload the recording, the Road Recorder application gets the authorization token from the Routa application, see Section 5.6.

[26] https://lockito-app.com/

5.4. Road Recorder management application

Road Recorder management application is the Web frontend application for the supervisor user of the system. The data and contents created by the Road Recorder are monitored and managed with the Road Recorder management application. The Road Recorder management application is developed with Vue.js[27] (version 2.6), a popular JavaScript framework.

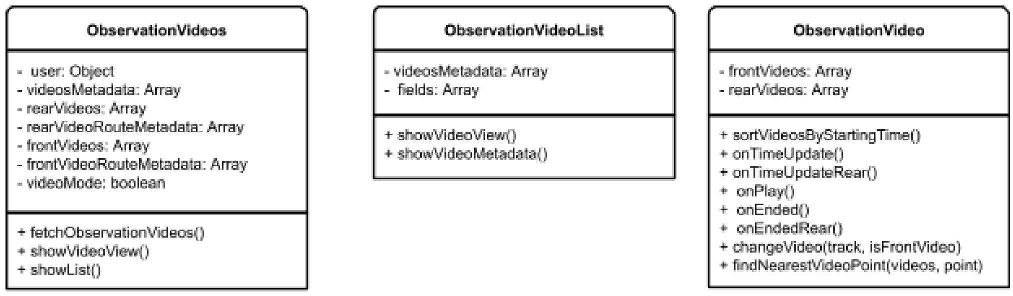

Figure 5.7. Class diagram of Road Recorder management application.

Figure 5.7 presents the class diagram of the Road Recorder management application. ObservationVideos class handles the video and recording metadata fetching and pre-processing. It also facilitates search functionality. ObservationVideoList class shows the list of the searched videos. Users can select a front and rear video from here to play in the video players. ObservationVideos class has two media players to play front and rear video supplied from ObservationVideoList. Users can compare the videos and inspect videos in specific geo-location on the map.

The first GUI that users see in the Road Recorder management application is the search view. Figure 5.8 represents the final design of the search GUI. Company supervisors can search for recording data with different parameters. For example, videos can be filtered using a specific date, company, or job. Searched videos are listed and can be ordered using their attributes. The related customer name, contract, and work order of recordings are shown along with the recording start time. Each list item has a dropdown button used to select the video to play as the front or rear video of a road.

Selected rear and front videos of road fixing work can be played by using two media players placed side by side. The final design of the video playback screen is shown in figure 5.9. A google map is placed just below the players to track recorded vehicle locations on the road. Both the videos and locations are synchronized with one another. The map shows the specific location of the currently played video playback point. Also, a video can be played at a specific geo-location point. Observations are pinned on the map, which on click, shows observation details. A dropdown selector is added to the map to filter these observations. After watching the videos, the user can go back to the search list using the back arrow at the top of the screen.

[27] https://vuejs.org/

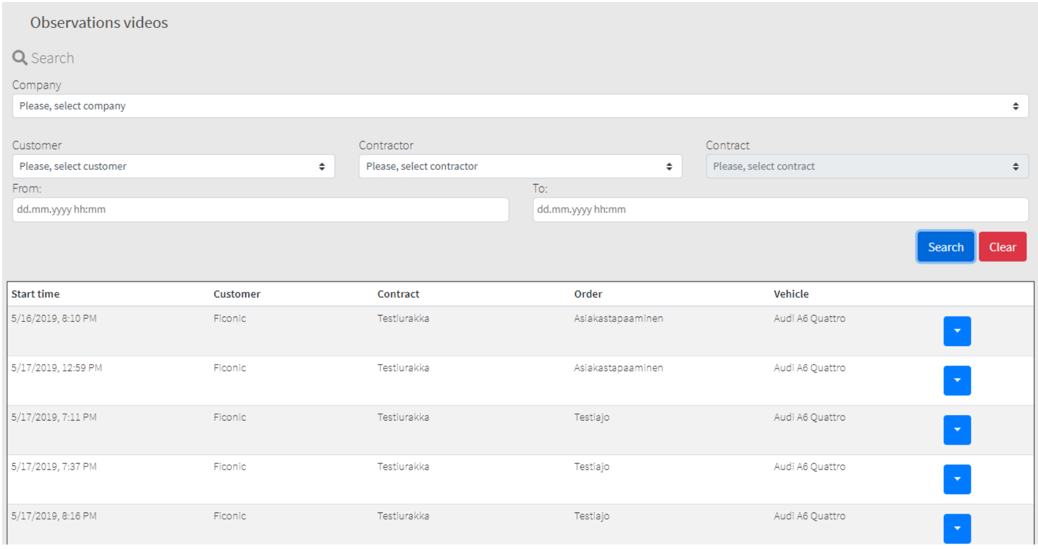

Figure 5.8. Video searching interface.

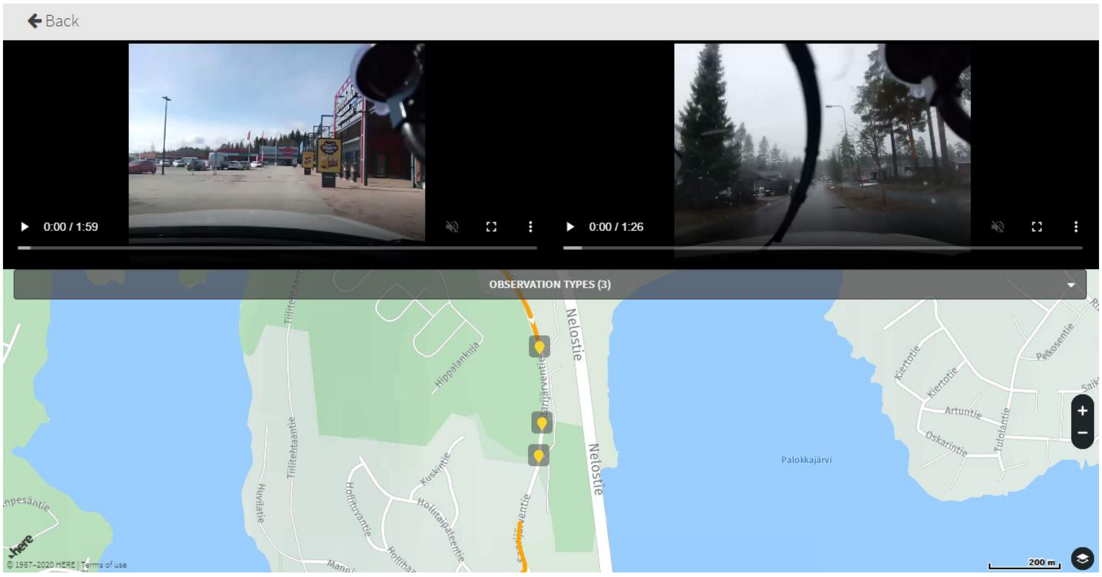

Figure 5.9. Video playback and comparing interface.

Different third-party frameworks and libraries were used in the implementation of frontend. Vue.js is chosen to implement the web frontend. Vue.js is chosen because it is lightweight, easy to learn, free to use, and has excellent community support. We used the licensed version of the HERE WeGo[28] map for location visualization. They have well-documented JavaScript APIs to support development.

[28] https://wego.here.com

6. EVALUATION

The Road Recorder Android and Road Recorder Management applications have been evaluated with test users to find out their usability and understand if they meet the customer requirements and which aspects can be improved. The term usability defines itself as it indicates how much a product is usable [44]. Usability indicates the ease of use of a product. Even though the user group of Road Recorder applications is not diverse in the truest sense of the word, our target group has various people on a smaller scale, including technical and non-technical background users.

The evaluation has been done separately for the Road Recorder Android application and Road Recorder management application. We used the following approach for both the applications: First, we set the concrete questions we are interested in the application to get answers from the user study. After that, we studied the evaluation method and decided on the types of user tests that would be most suitable for our purpose. Then we decided who will be our test users. After that, we designed tests and interviews for both applications. Once all the materials were ready and approved, we conducted the study. We gathered the necessary data for the analysis, helping to understand what opportunities and implications exist in our applications.

6.1. Evaluation of Road Recorder Android application

Though the initial idea was to test the Road Recorder application with the actual drivers who work on the road, it was challenging to manage such users who want to participate. So, we changed the plan to do the tests with regular car drivers with Finnish driving licenses who have been driving at least for two years in Finland. We used the Wizard of OZ approach [45], where the user's reactions are observed in different situations. We forced different situations on the application while the user was performing tasks. Examples of situations include disconnecting Wi-Fi and stopping video recording. The observation was focused on how users behave in these situations and handle them.

6.1.1. User study procedure and setup

For the Road Recorder application evaluation, we defined the following five tasks to be performed by test users in the car, which were all given at once to the participants at the beginning of the test.

1. Start Road recorder Android application and connect with the Routa app.
2. Switch between the video preview and no-preview.
3. Check if a recording is going on.
4. Observe connection status
5. Find uploaded and pending recordings.

We decided not to conduct the tests while users are on the road for security and legislation issues. Instead, we used a parked car, keeping the engine running. There were 5 participants in the Road Recorder application test. Two of them are from technical backgrounds, and others have experience of daily life technology usages.

The users performed the tasks sitting on the driving seat, pretending that they are driving while doing the tasks. The device having the Road Recorder application was set at the left corner of the car's dashboard. In contrast, the device with the Routa Android application was set at the right side of the driving wheel, facilitating easy interaction (see figure 6.1). Two different cameras were used to record the reaction of the users with their consent [Appendix 1]. Tasks are given on a printed paper and explained briefly. Once all the tasks are done, users are given a Google form to fill up [Appendix 3] to give their anonymous biographical data and comments and opinions about the application. The tests took 15-20 minutes to perform. Each participant was offered a coffee ticket at the end of the test.

Figure 6.1. Device orientation for Road Recorder tests.

6.1.2. Results

Most of the study's answers were collected with Google forms due to its easy distribution and helpful data analysis tools and are displayed below in this chapter. Regarding the demographic data, all five participants were males, 4 of whom were in their thirties and only one in their twenties. Two of them were researchers in the universities, and three of them were students.

Participants were asked to comment on the questions related to each task. The Likert scale (Strongly disagree - Strongly agree) was used. The following table shows the percentage of answers on different scales.

Table 6.1. Percentage of answers on different scales

Question	Strongly Disagree (%)	Disagree (%)	Neither agree nor disagree (%)	Agree (%)	Strongly Agree (%)
It was easy to understand that the Routa and Road	0	0	0	40	60

Recorder applications are connected.					
The 'Connected' notification message was in the right position.	0	0	0	20	80
Video preview was enough to set the phone in the right position.	0	0	0	60	40
Recording status was indicated.	0	0	40	40	20
Connection status notifications were clear enough.	0	0	0	0	100
Notification messages distracted you.	60	0	20	0	20
It was easy to find the recording logs.	0	0	0	20	80

With the Likert scale, we also allowed participants to answer some free-form questions where they could answer about the app features.

For the question “Which features you liked the most and why?”, two of the participants mentioned the video preview. “Video preview and logs, as they are easily defined and straightforward to find”, wrote one of them. One participant liked the smooth auto connection between the Road Recorder and Routa application.

And for the question “Was there any feature you didn’t like?” two users said that there is no feature they did not like specifically. “Recording logs are not very useful. Thumbnail of the recordings could be shown to understand which recording it is”, suggested one of the participants.

Participants suggested some features to improve the usability of the system. For example, one of the users suggested using a more prominent display and combining both the Routa and Road recorder into one so that drivers do not need to take care of two devices. One user suggested adding a button to connect and disconnect applications manually to have better control of the connection. Another user wanted the feature to see the recording log details. He was trying to see the recording log details by clicking on a recording log item. "Video icon could be replaced with a recorder icon. Instead of showing 'Recording', it should say- 'Recording in the background', and the icons should have active and passive states to indicate which functionality the user is on " - suggested by one of them.

6.1.3. Analysis

From table 6.1, we see that most of the users either agreed or strongly agreed with the positive statements. For example, all of the participants agreed that the connection status notification was clear to them. Also, connecting between the Routa Android application and the Road Recorder Android application was easy for all the participants. The same applies to video preview and recording log functionalities. From this, we evaluated that these features do not need immediate improvement.

Some features can still be improved, for example, notification messages. There was at least one user who was distracted by notification messages. To 40 percent of the participants, the video recording status icon was neither clear nor unclear, indicating a chance to make it more straightforward.

From the feedback from participants of the study, we evaluated that some features can be improved as follows:

1. The video icon can be changed to a Recorder icon. Recording status messages can also be improved, for example, instead of showing “Recording” we can show “Recording in the background”. Icon and text color can be changed to active or passive depending on which view they are on.
2. The recording log view can be improved. Users should be able to see more details of a recorded log. They should be able to see a thumbnail of the video to understand which recording it is.
3. A button could be added to manually connect and disconnect the Road Recorder with the Routa application.
4. Change the position of the notification messages to reduce distraction.

We should admit that there were some limitations of this study:

1. We could not manage enough road maintenance users for the test, so the test was done with regular car drivers.
2. We could not find enough test users because of the Covid-19 pandemic.
3. The application could not be tested in a real driving situation because of the legislation and safety of the test users.
4. As users did not have to drive the car during the test, it was impossible to know how they would react in different situations in real-life applications.
5. Because some users were from an information technology background and knowledge about the modern application components, the results might be biased.
6. Test users were either students or University researchers. Moreover, gender and age groups were not diverse enough, which could lead to bias in results.

Overall, the study participants found that the Road Recorder Android application is easy to use as most of the features do not need many changes.

6.2. Evaluation of Road Recorder Management application

The Road Recorder management application is for the supervisor user of the Routa Systems. Though the final production version of the Road Recorder management will be tested with the supervisor users from the industry, for evaluation in this thesis, we recruited test users with different backgrounds. The test's target is to evaluate different application components and find the SUS score [46] of the application. The SUS score is an easy, quick, and most common way to understand the usability of an application. In the SUS evaluation process, the user first experiences the application and then participates in answering the SUS questionnaire.

6.2.1. User study procedure and setup

For the Road Recorder management application evaluation, we defined the following tasks:

1. Login to the Routa web app.
2. Go to the recording list from the observation menu.
3. Search video recording with a time range.
4. Select and play front and rear video together.
5. Check road conditions in a specific geo-point from the map.
6. Go back and play another video.
7. Log out from the system.

The tasks were also given on a printed paper at the beginning of the test and described briefly. There were also 5 participants in the Road Recorder management application test, of which 2 of them also participated in the Road Recorder application tests. The tasks took 10-15 minutes to finish. After completing the tests, users were given a Google form with SUS questions and some anonymous biographical and opinion questions. The test took about 20 minutes. Participants were also given coffee tickets to thank for their contribution.

6.2.2. Results

The SUS tests and the follow-up questions [Appendix 5] were asked using Google form for easy access and secure storage of the data. Among the five participants, one was female, and the rest were male. Four participants were in the 20-29 years old range, and one was in 30-40. The SUS questions and the SUS score for each user are presented in the table below.

Table 6.2. The SUS scores of Road Recorder management application

	Question	User 1	User 2	User 3	User 4	User 5
1	I think that I would like to use this system frequently	5	5	4	4	2

2	I found the system unnecessarily complex	1	1	4	3	4
3	I thought the system was easy to use	5	5	2	4	2
4	I think that I would need the support of a technical person to be able to use this website	2	2	1	1	4
5	I found the various functions in this system were well-integrated	5	4	3	3	3
6	I thought there was too much inconsistency in this system	1	1	4	2	5
7	I would imagine that most people would learn to use this system very quickly	4	5	4	4	4
8	I found the website very cumbersome to use	2	1	2	2	3
9	I felt very confident using the system	4	4	3	4	1
10	I needed to learn a lot of things before I could get going with this system	1	2	3	5	5
	SUS SCORE	**90**	**90**	**55**	**65**	**27.5**
Average SUS score = 65.5						

In the follow-up question section, we have got some critical suggestions from the test users. These will help to improve the usability of the application. For example, when asked about the feature that can be improved more, most test users mentioned that the video selection for playing in the players could be improved. Test users most liked the side-by-side video playing and comparing feature, while the most unpleasant feature was choosing and starting the video playback. We also got some small feature requests, for example, indicating which video is selected for the front player and which video is for the back, a play button that will play and pause both the videos at a time, detecting potholes, and roadblocks from the video and mark them.

6.2.3. Analysis

The System usability scale does not give an absolute truth but a measurable score of the functionalities to relate with other functionalities scores. It will tell us what to improve out of the measured functionalities. For reference, here is how the scores should be interpreted [47]:

A Score over 80: These systems have great usability.

A score under 70: These systems have average usability.

A score under 50: These systems have very poor usability.

Table 6.2 shows how our test users answered usability questionnaires. Based on those answers, we calculated the SUS score for our application (average is 65.5). It falls to the second category with average usability. The score indicates that there is room for improvement. The score of question 4 and 10 indicates that the learnability of the application is very good, but some of the participants found the system a bit complex and need further assistance.

Though the application wasn't unnecessarily complex to the participants, there were some improvements and features requested from the participants. Some of them are listed below.

1. Provide more metadata and a preview of the videos to help understand which video the user is selecting.
2. Coupling the front and rear video so that the user does not need to understand manually which video is a rear video for a front video.
3. Feedback to the user that the video is selected for front or back video.
4. A button to play both front and back video with one click.
5. Better organization of the video searching and selection menus.

Similarly, there were certain limitations of this study:

1. We could not manage enough real target industry users for the test, hence the test is done with test users of various backgrounds.
2. Because of the current Covid-19 pandemic situation, tests were done online. As a result, it was not possible to record users' reactions during different stages of the test.
3. As the Road Recorder Management application is a part of the bigger Routa web application, we could not get what would be the real usability score if the Road Recorder Management application is a stand-alone application.
4. Here also the result is biased by age group and gender as there were not enough test users of diverse ages and gender.

As can be seen from the study, the video choosing functionality is the one that needs critical improvements, Other than that, different menus and video playback functionalities do not need changes.

6.3. Evaluation against requirements

This section analyzed both the Road Recorder Android application and Road recorder management application against the requirements that we defined in chapter 4, section 4.1.

6.3.1. Road Recorder Android application

Functional requirements:

Table 6.3. Functional requirements evaluation of Routa Android Application

Requirement	Analysis
Wireless connection between host devices of Routa application and Road Recorder application.	Routa And Road Recorder Android applications are connected through the common Wi-Fi they are connected with.
Unobtrusive connection of the devices.	Devices connected in the background without user interaction.
Automatic reconnection of the applications and devices.	If devices disconnect for any reason, it keeps retrying to reconnect automatically and inform users.
Start and stop recording automatically.	Starting and stopping recording happens automatically depending on speed and work mode.
Record and store geo-location points as metadata of the videos.	From table 5.1 we see that the 'RECORDINGS' table stores various metadata of recordings along with geometry. Geometry consists of the latitude and longitude of the location.
Upload recorded video and metadata to the server automatically and asynchronously.	Recorded videos are uploaded automatically and asynchronously when network is available, which is described in the Implementation chapter, section 5.3.4
Track uploaded recordings, recordings in the queue, and upload failed recordings.	With the recording log user interfaces, users could browse video recording logs.
Watch recording display preview.	Video recording view and preview features are implemented.
Always keep the video focus on the road surface.	This could not be implemented properly due to the limitation of Android camera API 2.0. Camera API doesn't help to keep the focus on any specific distance. This can be improved in the future when the feature is added to the API.

Performance requirements:

Table 6.4. Performance requirements evaluation of Routa Android Application

Requirement	Analysis
Establishing the connection between Road Recorder and Routa application should be done within 3-5 seconds.	The connection happens within 1-5 seconds of starting both applications.
Starting and stopping a recording should be done within 1 to 2 seconds. Otherwise, it can cause missing spots in the videos, hence cause valuable data loss.	Starting and stopping recording is controlled by the Routa application, which sends a signal to the Road Recorder application when needed. The signal reaches within milliseconds which is quick enough for this case.
The application should be available for use on appropriate devices at any time and place.	The application is written on the native Android framework and can be installed on any Android device. Any user with a user id can use the application. The application is only available to test users now.
Automated testing should be implemented to reduce bugs.	Travis CI[29] is used to test continuous integration. However, there is no automated functional test implemented. The application has been tested manually across different devices.
The Road Recorder application should be compatible with Android 6 Marshmallow (API 23) and higher versions.	Initially, the minimum target API version was 23, the application is developed only for API version 27 or higher. So, the application does not meet this requirement.

User Interface Requirements:

Table 6.5. User interface requirements evaluation of Routa Android Application

Requirement	Analysis
The Road Recorder application interface should be compatible with all screen resolution sizes, including tablets and large screen mobile phones.	The application is built responsive, which means it will auto-scale with the screen size and resolution of the device.

[29] https://travis-ci.com

It should only support landscape/horizontal mode.	The application is locked to landscape mode and cannot be used in portrait mode.
The application should contain standard android interface design components to understand each icon's meaning easily.	The application follows the common android design component. There was one user who could not understand the recording icon which we will need to redesign.
Localization of the user interfaces.	The application provides user interfaces in English, Finnish, and French.

Altogether, the Routa Road Recorder application meets most of the requirements.

6.3.2. Road Recorder Management application

Functional requirements:

Table 6.6. Functional requirements evaluation of Road Recorder management application

Requirement	Analysis
Search videos with metadata (Address, Work contract, and order, etc.).	Video can be searched with the contract, work order, and date range which is enough at this stage.
Select and play front and rear videos side by side to compare road conditions.	Front and rear videos can be selected and play side by side with different media controls.
Ability to watch the video at a certain geo-point.	Video can be played/paused in a specific geo-point from a map.
Enable the location tracking on the map while playing a video.	Location progress is drawn while the video progresses in the player.
Automatic detection of road cracks on the video.	It has not been implemented in this version of the application.
Automatic detection of road fixes.	It has not been implemented in this version of the application.

Performance requirements:

Table 6.7. Performance requirements evaluation of Road Recorder management application

Requirement	Analysis
In the Road Recorder management application, searching and playing videos are demanding tasks for response time. Users should be notified of the progress.	Data has been fetched from the backend using lazy loading, and a loader is shown during the data load.
Feedback should be given to the users if the task is delayed for more than 5 seconds, or there is a failure.	Toast messages are shown to the user in case of failure.
The server and database should be scalable.	The server and database are hosted using the AWS elastic beanstalk[30] which takes care of auto-scaling of the server and database.
The database should be indexed to perform an efficient video search.	This version of the application database is not indexed, hence searching with large data is slow.
The Road Recorder management application should be compatible with any modern JavaScript-enabled browser.	The application is written in the Vue JS framework, hence compatible with a JavaScript-enabled browser.

User Interface Requirements:

Table 6.8. User interface requirements evaluation of Road Recorder management application

Requirement	Analysis
The Road Recorder management application should be easily resizable in the browser.	The application is developed using responsive CSS libraries, which enables easy resizing in different screen sizes.
Interfaces should be mobile browser-friendly, and responsive.	The application is customized for mobile browsers.

[30] https://aws.amazon.com/elasticbeanstalk

Standard error messages should be shown to the users.	Custom toast messages with different colors are used to show different errors, warnings, and notification messages.
Localization of the user interfaces.	The application provides user interfaces in English, Finnish, and French.

Security Requirements:

Table 6.9. Security requirements evaluation of Road Recorder Android and Road Recorder management application

Requirement	Analysis
User's passwords should be encrypted before saving in the storage	Passwords are encrypted using the sha256 encryption algorithm before saving them into the database.
Only authorized users with proper access rights should see the information of other users.	Role-based token authentication is used to authenticate users to access different resources.
Secure communication between the Web servers and other components.	HTTPS connection is used for both backend and frontend access which is the most secure communication protocol for servers and clients.
Proper security configuration at the server-side protecting from unauthorized access	All the resources are token protected which can be gained only by using a registered username and password, ensuring the unauthorized access block.

From the analysis, we can say that the application meets most of the defined requirements. Despite the low SUS score of the Road Recorder management application, it still fulfills most of its requirements.

7. DISCUSSION

In this thesis, the Road Recorder Android application and a web console to explore the content collected with the Road Recorder application have been developed and evaluated with real users.

The main objective of this thesis was to develop a smartphone-based in-vehicle application for road work inspection and a web tool for the management of the video and metadata collected through the smartphone application. The result of Road Recorder Android application tests indicates that the application can serve its purpose, which is gathering the road data and videos for inspection. On the other hand, the Road recorder management application also provides facilities for analyzing the collected data, but it needs more improvement in its user interfaces for better usability. Though the developed applications are not yet production-ready, they can be used as pilot programs in the production environment.

While the municipalities need to take photos of every road fix they have done, the Road recorder system removes this burden by taking and storing simple video and metadata of the road surface. Sometimes it is hard to find the road fix in the videos without extra effort. This could be improved by using some image processing algorithms and mobile sensor data to highlight the road anomalies on the video. For instance, Yan et al. [50] presented a prototype of a video-based pavement distress screening (VPADS) system using Canny Edge detection [51], Boundary contour analysis [52], and probabilistic Hough transform algorithm [53], which can detect road anomalies in video. Another example is the work by Andrew et al. [54], who presented and analyzed different schemes where they used smartphone sensor data of crowd users to identify road anomalies. Integrating one of these approaches could significantly improve the user experiences of Road Recorder.

One of the challenges during the development phase was to test the Road Recorder Android application indoor. GPS sensors are not yet developed enough to give precise indoor location. The location simulator helped to overcome this by simulating location and speed changes. Greenhill et al. [11] tested a mobile surveillance system in an actual road situation but did not mention how it was tested indoors during the development stage. Gao J. et al. [48] suggested different approaches to test android mobility applications, including using a simulator application like Routa used, with a preconfigured location in the actual device, and using other users' devices.

Conducting user studies also faced some challenges. Particularly, finding the test users for testing the Road Recorder application was hard during the Covid-19 pandemic. It is an exceptional situation related work never faced. Conducting the user study for the Road Recorder management application was easier to organize because it was possible to conduct it online.

Also, the Road Recorder testing was done in a controlled environment. It allows observing different situations while testing the application closely. However, it sometimes cannot present the actual context of the mobile application uses [55]. For example, in the Road Recorder testing, connectivity was smooth throughout the testing, which could be different in an actual work situation. Zhang et al. [55] suggested a field study approach for such testing where the users will physically move around in a dynamically changing environment resulting in changing test parameters worth trying in the next round of evaluation.

As the vehicle was stationary during the test and provided explicit surveillance videos, the resulting application from this thesis indicates that this kind of smartphone application could have potential in the surveillance industry, not just in the in-vehicle system.

8. FUTURE WORK

In this thesis, we have evaluated the first version of the Road Recorder application. Doing any more future work on this project depends on the client's requirements and budget, but there is room for future improvement and enhancement in the application. Chapter 6, sections 6.1.2 and 6.2.2, have described some improvements and feature requests from the test users. In the next version of the application, those can be implemented.

In the Road Recorder Android application, we can add a media player to preview the recorded video and add a description to identify the videos better later. The video recording function can be improved to keep the video focus continually on the road when there is an updated camera API from the Android framework.

In the Road Recorder management application, the overall video searching and playing functionality can be improved. We can implement some video processing algorithms to compare frame by frame of the videos and mark different road conditions in the video in future work. Automatic detection of road surface anomalies using AI and machine vision can also be implemented in the future.

To overcome the limitation of the previous user tests, both the applications should be re-evaluated with some real users in the actual driving situation. All in all, for both applications, overall usability can be improved. As improving usability is an iterative process, the application should go through a few more rounds of improvements and evaluations. Also, the applications should be tested and evaluated with customers to understand the usability in the industry and see how much of the expectations are met. After conducting the user tests, we realized that we could ask users to rate each task individually, which could help us better understand each task's usability. It can be done in the next interaction of the tests.

9. CONCLUSION

Nowadays, every sector of human life is influenced by the use of smartphones. Professional work is handled and monitored using different smartphone applications in almost every sector, like real-time traffic information, postal services, hotel industries, employee management systems. A similar situation is with the use of smartphones in road construction works.

While most mobile surveillance systems use separate camera modules for in-vehicle surveillance systems and then process those in a different computing system [2] [11], the Road Recorder system, developed in this thesis, uses smartphones to record and process the videos. It can help to build low-cost commercial versions of such kinds of systems. Moreover, using smartphones makes our system highly portable and easy to set up in any vehicle.

After this thesis, Ficonic Solutions Oy now has a new software product. With some improvements, this can bring a significant number of clients to the company. Particularly, the thesis is vital for the clients of Routa Systems. With the use of the Road recorder tools, they would reduce the manual work proofing. This thesis was significant for the writer because he got the chance to learn some new technologies by developing the applications and learned HCI aspects in more detail.

Road recorder application has potential in different road and construction worksites with zero or minor changes in its functionalities. The application can reduce the amount of manual work in a worksite where constant monitoring and proof of work need to be stored, hence saving time, money, and resources. With the advancement in image processing and object detection, the Road Recorder can be used in more use cases in the future.

10. REFERENCES

[1] Navon, R. (2007). Research in automated measurement of project performance indicators. Automation in Construction, 16(2), 176-188.

[2] Ibrahim, Y. M., Lukins, T. C., Zhang, X., Trucco, E., & Kaka, A. P. (2009). Towards automated progress assessment of workpackage components in construction projects using computer vision. Advanced Engineering Informatics, 23(1), 93-103.

[3] Freimuth, Henk, and Markus König. "Planning and executing construction inspections with unmanned aerial vehicles." Automation in Construction 96 (2018): 540-553.

[4] Quater, P. B., Grimaccia, F., Leva, S., Mussetta, M., & Aghaei, M. (2014). Light Unmanned Aerial Vehicles (UAVs) for cooperative inspection of PV plants. IEEE Journal of Photovoltaics, 4(4), 1107-1113.

[5] Hallermann, N., & Morgenthal, G. (2014, July). Visual inspection strategies for large bridges using Unmanned Aerial Vehicles (UAV). In Proc. of 7th IABMAS, International Conference on Bridge Maintenance, Safety and Management (pp. 661-667).

[6] Rohlfing, T., Noring, J., Lancaster, J., Tree, E., & Hartsfield, A. (2006). U.S. Patent Application No. 11/325,204.

[7] https://www.ies.be/files/ADVISE-Aosta_12_DIMOU_video_surveillance_systems.pdf

[8] Marcenaro, L., Oberti, F., & Regazzoni, C. S. (2000, September). Change detection methods for automatic scene analysis by using mobile surveillance cameras. In Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101) (Vol. 1, pp. 244-247). IEEE.

[9] Kitahara, I. (2008, October). Interactive video surveillance by using environmental and mobile cameras. In 2008 World Automation Congress (pp. 1-6). IEEE.

[10] Jackson, S., Miranda-Moreno, L. F., St-Aubin, P., & Saunier, N. (2013). Flexible, mobile video camera system and open source video analysis software for road safety and behavioral analysis. Transportation research record, 2365(1), 90-98.

[11] Greenhill, S., & Venkatesh, S. (2006, October). Virtual observers in a mobile surveillance system. In Proceedings of the 14th ACM international conference on Multimedia (pp. 579-588).

[12] Chai, Y., & Wang, J. (2018, May). Design and Realization of Mobile Video Surveillance Car Based on ARM9 and Linux Platform. In 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) (pp. 1017-1021). IEEE.

[13] Ma, L., & Li, Z. (2017, October). The mobile video monitoring system based on Android. In 2017 Chinese Automation Congress (CAC) (pp. 6680-6684). IEEE.

[14] Pang, H., Jiang, L., Yang, L., & Yue, K. (2010, June). Research of android smart phone surveillance system. In 2010 International Conference On Computer Design and Applications (Vol. 2, pp. V2-373). IEEE.

[15] Abdelbacet, G., Ghada, Z., Mounir, S., & Abdennaceur, K. (2015, March). A real time environmental monitoring for smart city surveillance based GUI on Android platform. In 2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15) (pp. 1-6). IEEE.

[16] Tangtisanon, P. (2014, October). Android-based surveillance car. In TENCON 2014-2014 IEEE Region 10 Conference (pp. 1-4). IEEE.

[17] Sohn, S., Lee, H., Jung, H., & Kim, J. (2013, January). High-definition video-based multi-channel top-view vehicle surrounding monitoring system for mobile navigation devices. In 2013 IEEE International Conference on Consumer Electronics (ICCE) (pp. 655-656). IEEE.

[18] Okamoto, S., Nakagawa, M., & Morimura, A. (2005). U.S. Patent No. 6,911,997. Washington, DC: U.S. Patent and Trademark Office.

[19] Jang, J., Smyth, A. W., Yang, Y., & Cavalcanti, D. (2015, April). Road surface condition monitoring via multiple sensor-equipped vehicles. In 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 43-44). IEEE.

[20] Quintana, M., Torres, J., & Menéndez, J. M. (2015). A simplified computer vision system for road surface inspection and maintenance. IEEE Transactions on Intelligent Transportation Systems, 17(3), 608-619.

[21] Sy, N. T., Avila, M., Begot, S., & Bardet, J. C. (2008, May). Detection of defects in road surface by a vision system. In MELECON 2008-The 14th IEEE Mediterranean Electrotechnical Conference (pp. 847-851). IEEE.

[22] J. Yoon, B. Noble, M. Liu, "Surface street traffic estimation", MobiSys 07, pp. 220-232, 2007.

[23] R. Sen, P. Siriah, B. Raman, "Roadsoundsense: Acoustic sensing based road congestion monitoring in developing regions", SECON, pp. 125-133, 2011.

[24] Bhoraskar, R., Vankadhara, N., Raman, B., & Kulkarni, P. (2012). Wolverine: Traffic and road condition estimation using smartphone sensors. Fourth International Conference on Communication Systems and Networks (COMSNETS), January.

[25] Vittorio, A., Rosolino, V., Teresa, I., Vittoria, C. M., & Vincenzo, P. G. (2014). Automated sensing system for monitoring of road surface quality by mobile devices. Procedia-Social and Behavioral Sciences, 111, 242-251.

[26] Chien, T. Y., & Chung, S. L. (2014, July). Android-based driving assistant for lane detection and departure warning. In Proceedings of the 33rd Chinese Control Conference (pp. 174-179). IEEE.

[27] Taniguchi, Y., Nishii, K., & Hisamatsu, H. (2015, June). Evaluation of a bicycle-mounted ultrasonic distance sensor for monitoring road surface condition. In 2015 7th International Conference on Computational Intelligence, Communication Systems and Networks (pp. 31-34). IEEE.

[28] Jheng, Y. J., Yen, Y. H., & Sun, T. Y. (2015, June). A symmetry-based forward vehicle detection and collision warning system on android smartphone. In 2015 IEEE International Conference on Consumer Electronics-Taiwan (pp. 212-213). IEEE.

[29] Manoharan, R., & Chandrakala, S. (2015, February). Android OpenCV based effective driver fatigue and distraction monitoring system. In 2015 International Conference on Computing and Communications Technologies (ICCCT) (pp. 262-266). IEEE.

[30] Ghose, A., Biswas, P., Bhaumik, C., Sharma, M., Pal, A., & Jha, A. (2012, March). Road condition monitoring and alert application: Using in-vehicle smartphone as internet-connected sensor. In 2012 IEEE international conference on pervasive computing and communications workshops (pp. 489-491). IEEE.

[31] Kaner, Cem (November 17, 2006). Exploratory Testing (PDF). Quality Assurance Institute Worldwide Annual Software Testing Conference. Orlando, FL. Retrieved November 22, 2014.

[32] Khan, M. E. (2011). Different approaches to black box testing technique for finding errors. International Journal of Software Engineering & Applications, 2(4), 31.

[33] Nidhra, S., & Dondeti, J. (2012). Black box and white box testing techniques-a literature review. International Journal of Embedded Systems and Applications (IJESA), 2(2), 29-50.

[34] Acharya, S., & Pandya, V. (2012). Bridge between Black Box and White Box–Gray Box Testing Technique. International Journal of Electronics and Computer Science Engineering, 2(1), 175-185.

[35] Winner, H., Hakuli, S., Lotz, F., & Singer, C. (Eds.). (2016). Handbook of driver assistance systems: Basic information, components and systems for active safety and comfort. Springer International Publishing.

[36] Coutaz, J. (1994, May). Evaluation techniques: Exploring the intersection of HCI and software engineering. In Workshop on Software Engineering and Human-Computer Interaction (pp. 35-48). Springer, Berlin, Heidelberg.

[37] Buse, R. P., Sadowski, C., & Weimer, W. (2011, October). Benefits and barriers of user evaluation in software engineering research. In Proceedings of the 2011 ACM international conference on Object oriented programming systems languages and applications (pp. 643-656).

[38] Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. Intl. Journal of Human–Computer Interaction, 24(6), 574-594.

[39] Dow, S., MacIntyre, B., Lee, J., Oezbek, C., Bolter, J. D., & Gandy, M. (2005). Wizard of Oz support throughout an iterative design process. IEEE Pervasive Computing, 4(4), 18-26.

[40] V-Model (2013) Available at: <http://www.cio.bund.de/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html>. [Accessed 17 April 2021

[41] Masuda, S. (2017, March). Software testing design techniques used in automated vehicle simulations. In 2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW) (pp. 300-303). IEEE.

[42] Expat-finland.com. n.d. [online] Available at: <https://www.expat-finland.com/pdf/driving_in_finland.pdf> [Accessed 17 April 2021].

[43] Iso.org. 2011. [online] Available at: <https://www.iso.org/obp/ui/#iso:std:iso:26262:-1:ed-1:v1:en> [Accessed 17 April 2021].

[44] Bevana, N., Kirakowskib, J., & Maissela, J. (1991, September). What is usability. In Proceedings of the 4th International Conference on HCI.

[45] Dahlbäck, N., Jönsson, A., & Ahrenberg, L. (1993, February). Wizard of Oz studies: why and how. In Proceedings of the 1st international conference on Intelligent user interfaces (pp. 193-200).

[46] Benoît, C., Grice, M., & Hazan, V. (1996). The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using Semantically Unpredictable Sentences. Speech communication, 18(4), 381-392.

[47] Nathan Thomas (2015), How To Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website.

[48] Gao, J., Bai, X., Tsai, W. T., & Uehara, T. (2014). Mobile application testing: a tutorial. Computer, 47(2), 46-55.

[49] Harwood, F., 2018. Wizard of Oz testing – a method of testing a system that does not yet exist. - Usability Testing & Market Research by SimpleUsability Behavioural Research Consultancy. [online] Usability Testing & Market Research by SimpleUsability Behavioural Research Consultancy. Available at: <https://www.simpleusability.com/inspiration/2018/08/wizard-of-oz-testing-a-method-of-testing-a-system-that-does-not-yet-exist/> [Accessed 25 May 2021]

[50] Yan, W. Y., & Yuan, X. X. (2018). A low-cost video-based pavement distress screening system for low-volume roads. Journal of Intelligent Transportation Systems, 22(5), 376-389.

[51] Canny, J. (1986). A computational approachto edge detection. IEEE Transactions on Pattern A

[52] Suzuki, S., et al. (1985). Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1), 32–46

[53] Kiryati, N., Eldar, Y., & Bruckstein, A. M. (1991). A probabilistic hough transform. Pattern recognition, 24(4), 303–316

[54] Fox, A., Kumar, B. V., Chen, J., & Bai, F. (2015, June). Crowdsourcing undersampled vehicular sensor data for pothole detection. In 2015 12th annual IEEE international conference on sensing, communication, and networking (SECON) (pp. 515-523). IEEE.

[55] Zhang, D., & Adipat, B. (2005). Challenges, methodologies, and issues in the usability testing of mobile applications. International journal of human-computer interaction, 18(3), 293-308.

11. APPENDICES

Appendix 1. Consent form for video recording of the Road Recorder Android application testing.

Appendix 1. Road Recorder testing plan with drivers

Appendix 2. Follow up questions for Routa drivers

Appendix 3. Road Recorder Management application testing plan with managers

Appendix 4. SUS questionnaire with follow up questions

Appendix 1. Consent form for video recording of the Road Recorder Android application testing.

Consent for video recording of the Road Recorder Android application testing.

Dear participant,

Your video will be recorded using multiple cameras while doing the Road Recorder Android application testing. The videos will be used to analyze the application and its usability. The videos will be available only to me and will be discarded after the analysis. The video recordings will be used only for the thesis. By agreeing with this, you are giving the consent to record the videos of you doing the tasks and using them in the thesis.

Participant number

Your answer

- ○ Agree
- ○ Not agree

Date

Date

mm/dd/yyyy

Place

Your answer

Appendix 2. Road Recorder testing plan with drivers

Testing plan with Drivers:

Target users: Users of Video Recorder Android application**.** Test users are the people with real driving experience on the road.

Target of the test: The target of this test is to evaluate the Road recorder application components. This test will help to evaluate and improve the app UI, notification system, and error messages. Some demographic data such as the age of the users will be collected. With this user test, we want to find the learnability and effectiveness, efficiency factors of usability. When users do the task, we are interested to know how quickly they are learning to use the system. After they learned to use the system, we want to know how accurate they are in doing the task and how complete is the task. Once users have done the task right, next we are interested to find how fast they are in doing the task.

Number of users: 5-7

Test method: Wizard of OZ approach. We will force show different situations on the app and ask to do some tasks. We will then observe and record how users behave in different situations.

Test setup: For the driver's test, a stationary parked car will be used. Users will be behind the wheel pretending that they are driving. Two cameras will be used to capture users' interactions and reactions with the system while doing the tasks. A printed instruction and the task list will be given to the users along with some vocal descriptions. Two Android devices will be used for the tasks. |One will be mounted at the middle top of the vehicle with the Road Recorder installed in it. Another device with the Routa application will be mounted near the steering.

Time needed: 20-30 min max

Planned tasks:

Tasks	**Scenario**	**Task for the user**	**Task for the observer**	**Follow up questions**
1	The user wants to connect between the Routa app and the Road recorder app to start the video recording. Routa app is up and running. Now it needs to connect the road recorder	Start Road recorder and connect with Routa app	Describe how two apps connect and observe if the user is doing it right. Observe what is done right or wrong	1. Was it easy to understand that apps are connected? 2. Is the connected message in the right position?
2	The user wants to see the video preview so that he can set the mobile in the correct position.	Switch between preview and no-preview	Observe if it is easy to switch preview.	1. Is preview enough to set the phone in the right position?

				2. Does the button indicate a preview?
3	The user needs to know if a recording is going on	See if a recording is going on	Start recording and observe if the user understand the recording condition	1. Is the recording status indicated?
4	Wi-Fi connection is lost	User should be notified that Wi-Fi is not connected	Observe if the user understand that no Wi-Fi is connected	1. Is the notification clear enough? 2. Was it distracting other activities?
5	It needs to be checked if all the recording log is sent to the server.	Go to the recording log menu and see uploaded and pending recordings.	Observe if it is easy for a user to find the log	1. Is it easy to find the logs?

Appendix 3. Follow up questions for Routa drivers

Follow up questions for Routa drivers

Thank you for volunteering to evaluate Road Recorder application. This questionnaire is to evaluate the Road Recorder application. It would take about 4 - 5 minutes to fill this questionnaire and the data is collected solely for the purpose of the thesis study.

Please fill this questionnaire after using the system to accomplish the tasks specified by the interviewer.

* Required

It was easy to understand that the Routa and Road Recorder applications are connected.

- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

'Connected' notification message was in the right position.

- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

Video preview was enough to set the phone in the right position.

- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

Recording status was clearly indicated.

- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

Connection status notifications were clear enough.

- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

Notification messages distracted you.

- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

It was easy to find the recording logs.

- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

Age *

Your answer

Gender *

- Female
- Male
- Other

Which features can be improved more?

Your answer

Which features you liked the most and why?

Your answer

Was there any feature you didn't like? why?

Your answer

Do you have any feature suggestions?

Your answer

Do you allow us to quote your comments in our thesis? *

- [ ] Yes
- [ ] No

Submit

Appendix 4. Road Recorder Management application testing plan with managers

Testing plan with Company Managers:

Target users: Company manager who will use the Video recording management web application. Test users are people who do not have a technical background.

Target of the test: The target of this test is to evaluate the Video recording management web application components and find the SUS score of the web console and improve the UI and functionality accordingly.

Number of users: 5

Test method: SUS evaluation approach will be used here. Users will be given a series of tasks to do with the system and then they will participate in a SUS test.

Test setup: Manager's test will be done using a windows browser. A brief description of the system and its functionality will be given vocally. Users will be then given a printed task list to complete the scenario presented below to follow. The tasks are orientated in a way that the participants will need to go through almost all the functionality of the application. User's behavior and execution time will be observed while they are doing the task to analyze later.

Scenario:
Road workers have done some road crack fixing work. They used the Road Recorder application to keep evidence of the road fix. The company manager now wants to check if the job is done properly. He knows when the job is done. He needs to search the recorded videos and compare the video of the road track before and after the road work.

Time needed: 15-20 min

Planned tasks:

1. Login in Routa web app
2. Go to recording list from observation menu
3. Search video recording with a time range
4. Select and play front and rear video together
5. Check road condition in a specific geo-point from the map
6. Go back and play another video
7. Logout from the system

Appendix 5. SUS questionnaire with follow up questions

SUS Questionnaire

Thank you for volunteering to evaluate the Road Recorder management application. This questionnaire is to evaluate the Road Recorder management application. It would take about 4- 5 minutes to fill this questionnaire and the data is collected solely for the purpose of the thesis study.

Please fill this questionnaire after using the system to accomplish the tasks specified by the interviewer.

* Required

I think that I would like to use this system frequently *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I found the system unnecessarily complex *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I thought the system was easy to use *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I found the various functions in this system were well integrated *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I thought there was too much inconsistency in this system *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I would imagine that most people would learn to use this system very quickly *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I found the website very cumbersome to use

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I felt very confident using the system *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I needed to learn a lot of things before I could get going with this system *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Age *

Your answer

Gender *

Choose

Which features can be improved more?

Your answer

Which features you liked the most and why?

Your answer

Was there any feature you didn't like? why?

Your answer

Do you have any feature suggestions?

Your answer

Do you allow us to quote your comments in our thesis?

- [ ] Yes
- [ ] No

Submit